Mobile Edge Computing in Wireless Communication Networks: Design and Optimization

A thesis submitted for the degree of Doctor of Philosophy

(Ph.D.)

by

Xiaoyan Hu

Communications and Information Systems Research Group

Department of Electronic and Electrical Engineering

University College London (UCL)

September, 2020

I, Xiaoyan Hu, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Sign:

Date:

Abstract

This dissertation studies the design and optimization of applying mobile edge computing (MEC) in three kinds of advanced wireless networks, which is motivated by three non-trivial but not thoroughly studied topics in the existing MEC-related literature. First, we study the application of MEC in wireless powered cooperation-assisted systems. The technology of wireless power transfer (WPT) used at the access point (AP) is capable of providing sustainable energy supply for resource-limited user equipment (UEs) to support computation offloading, but also introduces the double-near-far effect into wireless powered communication networks (WPCNs). By leveraging cooperation among near-far users, the system performance can be highly improved through effectively suppressing the double-near-far effect in WPCNs. Then, we consider the application of MEC in the unmanned aerial vehicle (UAV)-assisted relaying systems to make better use of the flexible features of UAV as well as its computing resources. The adopted UAV not only acts as an MEC server to help compute UEs' offloaded tasks but also a relay to forward UEs' offloaded tasks to the AP, thus such kind of cooperation between the UAV and the AP can take the advantages of both sides so as to improve the system performance. Last, heterogeneous cellular networks (HetNets) with the coexistence of MEC and central cloud computing (CCC) are studied to show the complementary and promotional effects between MEC and CCC. The small base

stations (SBSs) empowered by edge clouds offer limited edge computing services for UEs, whereas the macro base station (MBS) provides high-performance CCC services for UEs via restricted multiple-input multiple-output (MIMO) backhauls to their associated SBSs. With further considering the case with massive MIMO backhauls, the system performance can be further improved while significantly reducing the computational complexity.

In the aforementioned three advanced MEC systems, we mainly focus on minimizing the energy consumption of the systems subject to proper latency constraints, due to the fact that energy consumption and latency are regarded as two important metrics for measuring the performance of MEC-related works. Effective optimization algorithms are proposed to solve the corresponding energy minimization problems, which are further validated by numerical results.

Impact Statement

This thesis contributes to the design and optimization of mobile edge computing (MEC) in advanced wireless communication networks, including the wireless powered cooperation-assisted systems, the unmanned aerial vehicle (UAV)-assisted relaying systems, and the heterogeneous cellular networks coexisting with the central cloud computing. The research of MEC is motivated by the unprecedentedly ever-growing mobile data generated by a large variety of mobile applications. In order to meet the demands of massive data processing, offloading the computation tasks to the cloud is an ideal offer for resource-limited mobile devices. Furthermore, MEC brings the cloud closer to the edge of the network and to the end users, which is promising to improve quality-of-service. Through integrating computing, storage, and networking resources with the access points, computation-intensive and latency-critical applications like unmanned driving and augmented reality can be hosted at the edge of the network.

The attractive advantages of MEC have drawn great attention from both the academia and industry. MEC has recently been standardized in a European Telecommunication Standards Institute (ETSI) Industry Specification Group (ISG), and it is also been recognized by the European 5G Infrastructure Public Private Partnership (5G PPP) as one of the key emerging technologies for 5G networks. However, taking full benefits of MEC still faces many challenges: *i)* Limited battery

supply for mobile devices makes it difficult for them to offload massive data to the MEC servers; *ii)* Weak connections for edge users may restrict their access to the MEC resources at the access points; *iii)* MEC cannot entirely replace the central cloud computing which makes it necessary to explore the cooperations with central clouds. The research in this thesis bridges the gap to tackle these challenges and provides pioneer optimization results from the perspective of academia. The technology of wireless power transfer (WPT) in MEC networks is capable to provide sustainable energy supply for end users, and the technologies of user cooperation and UAV assistance are shown that can significantly enhance the ability of MEC networks in dealing with computation-intensive and latency-critical tasks. Also, it is illustrated that MEC and central cloud computing are highly complementary and great benefits can be attained when utilizing them both.

The research in this thesis also provides theoretical supports for the development of MEC in the industry. The advanced MEC and communication technologies studied in this thesis are promising to be utilized in 5G networks. In fact, 5G and MEC are inextricably linked. 5G is capable to increase speeds by up to ten times that of 4G, whereas MEC reduces latency by bringing compute capabilities into the radio access network (RAN). It is known that Google announced the Global MEC (GMEC) strategy, where Google and telecommunication providers will offer unique applications and services running at the edge delivered via the 5G networks. Also, AWS and Verizon Communications announced a partnership that will bring the power of the world's leading cloud closer to mobile and connected devices at the edge of Verizon's 5G Ultra Wideband network.

Acknowledgements

I would like to express my sincere appreciation to many people who give me supports for completing this thesis.

First and foremost, I would like to give my greatest gratitude to my Ph.D. supervisor Prof. Kai-Kit Wong for his great supports, guidance, and encouragement. He always inspires me to explore new research areas and encourages me to meet the challenges in academics. I appreciate his insightful comments and revisions in every paper we have produced together. His valuable advice will play a life-long effect on me. I also would like to acknowledge Prof. Kai-Kit Wong, UCL, and the China Scholarship Council for the financial support during my Ph.D. study.

Then, I would like to thank my group 'UCL Wireless' led by Prof. Kai-Kit Wong, including Dr. Lifeng Wang, Dr. Jialing Liao, Dr. Yongxu Zhu, Dr. Arman Shojaeifard, Dr. Muhammad RA Khandaker, Dr. Raoul Guiazon, Dr. Alexander Okandeji, Mr. Emanuele Gruppi, Mr. Zhiyuan Chu, Miss Haizhe Liu, and Mr. Hadumanro Randani Malau. The time I spent at UCL with you is one of the most precious memories of my life. I would like to give my special thanks to Dr. Lifeng Wang for his sincere help and valuable advice in academics.

Moreover, I would like to show my appreciation to Prof. Christos Masouros and Prof. Chin-Pang Liu for attending my MPhil to Ph.D. transfer and providing valuable comments on my research works. I am also grateful to Prof. Justin Coon

and Dr. Ioannis Psaras for being examiners of my Ph.D. viva, and their insightful comments highly improved the quality of this thesis. I am also thankful to my colleagues in UCL, including but not limited to Dr. Fan Liu, Dr. Tongyang Xu, Dr. Zhongxiang Wei, Dr. Pingfan Song, Dr. Abdelhamid Salem, Dr. Jianjun Zhang, Dr. Aryan Kaushik, Mr. Yin Bi, Miss Jun Qian, Miss Nanchi Su, Miss Xiaoye Jin, Mr. Mohammad Abdullahi, and Mr. Iman Valiulahi.

Last but not least, I would like to express my most sincere and deepest gratitude to my parents, who give me unconditional love and supports. I am grateful for their understanding, encouragement, and supports on me to pursue my Ph.D. degree and chase my dream at UCL.

Contents

List of Symbols

$\mathbb{C}^{x\times 1}$	the space of $x \times 1$ complex vectors
$\mathbb{C}^{x\times y}$	the space of $x \times y$ complex matrices
$\mathbf{x}$	a vector with appropriate dimension (boldface lower case)
$\mathbf{X}$	a matrix with appropriate dimension (boldface upper case)
$\mathbf{x}^T$ (or $\mathbf{X}^T$)	the transpose of vector $\mathbf{x}$ (or matrix $\mathbf{X}$)
$\mathbf{x}^\dagger$ (or $\mathbf{X}^\dagger$)	the conjugate of vector $\mathbf{x}$ (or matrix $\mathbf{X}$)
$\mathbf{x}^H$ (or $\mathbf{X}^H$)	the conjugate transpose of vector $\mathbf{x}$ (or matrix $\mathbf{X}$)
$\nabla_{\mathbf{X}} f(\mathbf{X})$	the Jacobian matrix of $f(\mathbf{X})$ with respect to (w.r.t.) $\mathbf{X}$
$\succeq$ (or $\succ$)	the component-wise inequality
a^+ or $[a]^+$	$\max\{0, a\}$ where a represents a real number or expression
$\Re\{\cdot\}$	the real-value operator
$\mathcal{O}$	the big O notation for showing the algorithm complexity
$\det(\mathbf{A})$	the determinant of square matrix $\mathbf{A}$
$\mathrm{tr}\{\mathbf{A}\}$	the trace of square matrix $\mathbf{A}$
$\mathrm{eig}\{\mathbf{A}\}$	the set of all the eigenvalues for square matrix $\mathbf{A}$
$\mathrm{eigvec}\{\cdot\}$	the eigenvector for a given eigenvalue
$\langle \mathbf{A}_1, \mathbf{A}_2 \rangle$	$\Re\{\mathrm{tr}(\mathbf{A}_1^H \mathbf{A}_2)\}$
κ	the processor's effective capacitance coefficient

$\{\theta_1, \theta_2\}$	parameters related to UAV's propulsion energy consumption
ν_k	the energy conversion efficiency for user equipment (UE) k
B	the system bandwidth
e	the base of the natural logarithm
E	energy in joule (J)
f	the CPU clock frequency
h_0	the channel power gain at a reference distance of d_0= 1m
H	the fixed altitude of the UAV
$[I, C, O]$	the computation task tuple
I	the size (in bits) of the computation task-input data
C	the CPU cycles required for computing 1-bit of task-input data
$O \in (0, 1)$	the ratio of task-output data size to that of the task-input data
K	the number of UEs
N_0	the average noise power at the receiver
P_0	the the access point (AP)'s energy transmit power
R	the achievable communication rate
$\mathbf{s}_k = (x_k, y_k)$	the horizontal locations of UE k
$\mathbf{s}_0 = (x_0, y_0)$	the horizontal location of the access point
t	latency in second (s)
T	the block length/the total task completion time
$\mathbf{u}_{\mathrm{I}} = (x_{\mathrm{I}}, y_{\mathrm{I}})$	the initial location of the UAV
$\mathbf{u}_{\mathrm{F}} = (x_{\mathrm{F}}, y_{\mathrm{F}})$	the final location of the UAV
$\mathbf{u}[n]$	UAV's location in each slot n
v	velocity in meter/second (m/s)
$\mathbf{v}[n]$	UAV's speed in each slot n

$V_{\max}$	the maximum available speed of the UAV
$W_0(z)e^{W_0(z)} = z$	the principal branch of the lambert W function

List of Abbreviations

1G	first generation
3D	three-dimensional
4C	communications, computing, control and content delivery
4G	fourth generation
5G	fifth generation
5GAA	5G Automotive Association
AMTE	average minimum transmit energy
AMTP	average minimum transmit power
AP	access point
APTEM	AP's transmit energy minimization
APTPM	AP's transmit power minimization
AWGN	additive white Gaussian noise
BS	base station
CCC	central cloud computing
CPU	central processing unit
CSI	channel state information
D2D	device-to-device
DNNs	deep neural networks

DVFS	dynamic voltage and frequency scaling
ETSI	European Telecommunication Standards Institute
FDD	frequency-division duplex
HetNet	heterogeneous cellular network
Hz	hertz
IaaS	infrastructure as a service
ICN	information-centric network
ICT	information and communication technology
IoT	Internet-of-things
ISG	Industry Specification Group
J	joule
KKT	Karush-Kuhn-Tucker
LoS	line-of-sight
LTE	long term evolution
m	meter
MBS	macro base station
MCC	mobile cloud computing
MEC	mobile edge computing
MIMO	multiple-input multiple-output
mMTC	massive machine-type communications
mmWave	millimeter wave
MRC	maximal-ratio combining
MTC	machine-type communications
NFV	network functions virtualization
NIST	National Institute of Standards and Technology

OFDMA	orthogonal frequency-division multiple-access
PaaS	platform as a service
QoS	quality of service
RAN	radio access network
RF	radio frequency
s	second
SaaS	software as a service
SBS	small base station
SCA	successive convex approximation
SDN	software-defined network
SES	sum-energy-saving
SESM	sum-energy-saving maximization
SINR	signal-to-interference-plus-noise ratio
SWIPT	simultaneous wireless information and power transfer
TDD	time-division duplex
TDMA	time-division multiple-access
UAS	unmanned aerial systems
UAV	unmanned aerial vehicle
UE	user equipment
V	volt
VM	virtual machine
V2X	vehicle-to-everything
W	watts
WPCN	wireless powered communication network
WPT	wireless power transfer

w.r.t.	with respect to
WSEC	weighted sum energy consumption
ZF	zero-forcing

List of Figures

Chapter 1

Introduction

1.1 Background

Cloud computing as an efficient computing platform have enjoyed rapid development over the last few decades, mainly driven by the ever-growing computing and processing demands of various client devices. Accompanied by the massive computing demands, higher quality requirements are also requested by users along with the astounding advances of communication and networking technologies. To this end, plenty of researchers and scientists have devoted to advanced techniques that can improve the efficiency and reduce the latency of computing services. Recently, a brightly new concept of mobile edge computing (MEC) has drawn great attention from both the academia and industry, which is promising to provide computing services with ultralow latency, high bandwidth, and real-time access, through shifting the cloud computing from the remote centralized data centers to the edge of mobile networks proximate to end users. In this section, we present the basic background of cloud computing, mobile cloud computing (MCC), and

MEC, and further the motivations and conditions that necessitate the shift of cloud computing from the central to the edge of the networks.

1.1.1 Cloud Computing: A Centralized Platform for Computing

The past few decades have witnessed the rapid advances of cloud computing as an emerging Internet-based technology which facilitates the online computing services for various users, including all sorts of organizations and personal devices. As defined by National Institute of Standards and Technology (NIST): Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1].

Cloud computing is a centralized platform, which is also known as central cloud computing (CCC), and the shared pool of resources is also referred as the data center or central cloud. The technology of cloud computing provides a promising way of increasing the capacity of infrastructures and reducing the overall cost through resource sharing, where the users can enjoy high quality of service (QoS) with minimum cost. Hence, the main purpose of cloud computing is to use the resources to the maximum level through centralized operations, combining them to achieve better performance and higher efficiency. The attractive features of cloud computing, such as scalability, inter-operability, feasibility, and pay-as-you go service principle, speed up its further development and integration with other advanced technologies.

1.1.2 Mobile Cloud Computing: Integrating Cloud Computing into Mobile Environment

The ever-growing number of mobile end-user devices along with the great amount of processing data have driven the rising of the mobile cloud computing (MCC). The technology of MCC integrates CCC into the mobile environment to facilitate mobile users taking full advantages of cloud resources [2–8]. Through offloading the computation data to the clouds, the computation tasks of mobile devices can be addressed by using resources at cloud providers other than the mobile devices themselves to host the execution of mobile applications. Such a cloud computing infrastructure where data storage and processing happen outside the mobile devices is specifically termed as 'mobile cloud', through which the cloud computing services can be accessed by the mobile user equipment (UEs) via the cellular core networks. Hence, the plentiful computing resources available at the clouds can be utilized to deliver elastic computing power and storage to support wide range of applications for the resource-limited mobile UEs. By migrating computational tasks from the UEs to the infrastructure-based cloud servers, MCC can improve the performance of mobile applications and reduce the energy consumption of UEs.

1.1.3 New Computing Challenges and Opportunities for 5G and Beyond Wireless Networks

It is well known that the latency is always a crucial performance metric for wireless services, no matter in the first generation (1G) or the fifth generation (5G) and beyond wireless networks. From 1G to the fourth generation (4G), the main target the wireless systems is the pursuit of increasingly higher wireless speeds to support

the service transition from voice-centric to multimedia-centric traffic with low latency. By leveraging the advanced 5G technologies, such as the massive multiple-input multiple-output (MIMO) and millimeter wave (mmWave) communications, it is capable of achieving the wireless speeds approaching the wireline counterparts. Hence, in light of the explosive evolution of information and communication technology (ICT) and Internet, the mission of 5G is much more complex and challenging beyond exploring higher transmission speed. Actually, 5G systems are expected to support services of communications, computing, control and content delivery (4C), and the latency requirements for all the 4C related services will become even more stringent.

Among the 4C services, the computing requirement will become a great challenge for 5G systems especially considering the explosively growing number of mobile and Internet-of-things (IoT) devices. In addition, a wide range of emerging mobile applications [9–13], from highly-interactive online gaming, virtual reality, to smart homes and automatic driving, etc., have unprecedentedly driven the increasing computing demands of UEs. One major characteristic of these applications is that they require intensive computations, which should be accomplished with low latency. Such computationally intensive applications easily exceed the ability of resource-limited UEs, not to mention the fact that they will drain their power quickly. Under this circumstance, a promising way to liberate the resources-limited devices from heavy computation workloads is to rely on external computing resources, either resorting to MCC or exploiting the computing resources at the edge of the mobile networks, e.g., MEC.

Although MCC is capable of providing cloud computing services for UEs, there exists one inherent drawback, i.e., the infrastructure-based central cloud

servers are usually located far away from UEs. Hence, accessing the MCC services induces excessive transmission latency, which highly aggravates the backhaul congestion. Besides, it is easy to encounter the performance bottleneck considering the finite backhaul capacity and the exponentially growing mobile data, and thus the computation offloading efficiency and user experience through MCC may severely degrade. Recently, more and more attention has been drawn to the the opportunities provided by MEC due to its proximity to end users.

As we mentioned before, the unprecedentedly growing number of edge devices, such as laptops, tablets, smartphones, and various wearable and sensor devices will bring great challenges for 5G wireless networks since these devices may require massive computing resources for operating application tasks which may beyond their own abilities. However, the densely deployed devices also provide some opportunities for facilitating edge computing. At every time instant, a large number of edge devices will be idle, and thus their available computing and storage resources can be harvested as a edge computing pool to support the devices with resource deficits. Besides the ultra-dense user devices, a great number of wireless access points (APs) will also be deployed to provide better coverage and higher QoS in 5G networks. A more typical mode of MEC is that a powerful computing server will be installed at each of the wireless AP, such as the small-cell base stations (BSs), gateways, Wi-Fi routers, etc., which can be easily accessed by the cellular connected or Wi-Fi connected mobile and IoT devices. The corresponding computing servers are referred as MEC servers with certain degree of cloud computing capabilities, and also known as edge clouds. This kind of MEC mode is what we are focusing on in this thesis. In a word, a variety of computing opportunities can be explored at the edge of 5G wireless networks.

1.1.4 Mobile Edge Computing With Clouds Shifting from the Central to the Edge

The explosion of demanding applications as well as the inherent drawback of MCC necessitate the shift of the cloud computing services from the remote data centers (central clouds) to the edge of the mobile networks, i.e., edge clouds, within the radio access networks (RANs). This brightly new kind of computing mode is well known as MEC, which exploits a new type of unified telecommunication and micro-datacenter node able to jointly provide networking, local processing, and storage resources for the support of novel 5G applications, such as IoT, vehicle-to-everything (V2X), machine-type communications (MTC), and immersive media, etc. Taking the applications of IoT as an example, MEC is a powerful computing paradigm that can assist in providing ideal services for IoT devices. As a distributed computing infrastructure, MEC is capable of bringing the computing capabilities close to the distributed IoT devices. In addition, deploying a number of edge computing nodes/servers in the IoT networks can locally collect, classify, and analyze the raw IoT data streams by local executions, rather than transmitting them to the cental clouds, which can significantly alleviate the traffic in the core networks and potentially speed up the IoT big data processing and improve the user experience.

In other words, MEC promotes to use cloud-computing facilities at the edge of mobile networks by integrating MEC servers at the wireless APs. This paradigm of computation offloading is motivated by proximity, ultralow latency, high bandwidth, and real-time access to radio network information, which is widely considered as an effective means to liberate the resource-limited UEs from heavy computation

workloads, e.g., [14–16]. With proximate access and distributed architectures, MEC is well known as a promising complementary counterpart of centralized cloud computing. In fact, MEC as one of the key enablers to shape the future advanced wireless networks has recently been standardized in a European Telecommunication Standards Institute (ETSI) Industry Specification Group (ISG) [17–19].[1]

1.2 Research Motivations

This thesis focuses on the design and optimization of MEC in three advanced wireless communication networks, which is motivated by the following three non-trivial but not thoroughly studied topics in the existing MEC-related literature.

- Recently, MEC has been widely used in cellular networks, focusing on improving the energy efficiency or reducing the latency of various cellular-based MEC systems [20–41]. In order to further task the full benefits of powerful computational resources at the edges and overcome the energy-limited drawbacks of traditional battery-based mobile devices, the technology of wireless power transfer (WPT) has been considered as an important paradigm to provide genuine sustainability for mobile communications [42–51]. Particularly, the form of wireless powered communication network (WPCN) is utilized to achieve the synergy of integrating MEC with WPT [52–55]. However, the existing wireless powered MEC works do not carefully envisage the terrible fact that WPCNs are susceptible to suffering from the so-called "double-near-far" effect, which occurs because the farther UEs from an AP harvest less energy and are also required to communicate in longer distances [47–49]. To effectively resist the double-near-far effect in wireless powered

[1] More details of MEC and the related literature review are given in Chapter 2.

MEC networks and improve the system performance, the technology of user cooperation can be leveraged as a promising solution.

- The attractive advantages of unmanned aerial vehicles (UAVs), such as easy deployment, flexible movement, and line-of-sight (LoS) connections, etc., have driven the extensive research on UAV-enabled wireless communications in recent years [56–62]. Moreover, it is a great attempt to leverage the technology of UAV in MEC systems, where the special features of UAV are promising to achieve extra performance improvement [63–68]. Nevertheless, the existing MEC works concentrate either on the cellular-based MEC networks or the UAV-enabled MEC architectures, where only the computing resources at the APs or at the UAV processing servers are utilized. In fact, it is risky to rely solely on the APs or the UAVs to complete UEs' computation-intensive latency-critical tasks, considering the facts that the UEs' wireless fading channels accessing to the APs may be severely degraded and the limited computing capabilities of the UAVs may be incapable of dealing with UEs' computation tasks. Hence, jointly leveraging the advantages of cellular-based and the UAV-enabled MEC architectures, and considering a UAV-assisted MEC system with cooperation between UAV and AP can make a difference.

- Even though MEC has been regarded as a promising trend to deal with the ever-growing mobile computing data, it cannot entirely replace the present central cloud computing, due to the fact that edge computing is set to push limited processing and storage capabilities at the APs close to UEs but may be incapable of dealing with big data processing. For UEs with highly computation-intensive tasks, the edge computing servers/clouds may be

incapable of providing them with satisfactory computing services. Under this situation, CCC/MCC has been shown to be an effective solution. The latest white paper from ETSI has further illustrated that central cloud computing and edge computing are highly complementary and significant benefits can be attained when utilizing them both [69]. However, the architecture with the coexistence of edge and central clouds has not been thoroughly studied, especially from the perspective of communications [14]. In conclusion, a heterogeneous architecture consisting of both the edge servers at the small BSs (SBSs) and central clouds connected to the macro BS (MBS) can not only make up the drawbacks of MEC and MCC but also improve system performance as well as user experience.

1.3 Thesis Organization and Main Contributions

Sequential to this chapter of introduction, the rest of this thesis is organized as follows. Chapter 2 introduces some fundamental concepts and the related state-of-the-art works. Driven by the three research motivations shown in Section 1.2, we construct three technical chapters sequentially to deal with the problems derived from the motivations, respectively in Chapter 3, Chapter 4, and Chapter 5. The conclusions of this thesis are summarized in Chapter 6. And then Chapter 7 presents the future works based on this thesis. Figure 1.1 shows the architecture of the thesis organization. The content and contributions of the chapters following the Introduction are summarized as follows.

Chapter 2: Fundamental Concepts and State-of-the-Art Works. In this chapter, we present the fundamental concepts used in this thesis, and a comprehensive literature review is also given to demonstrate the relevant state-of-the-art

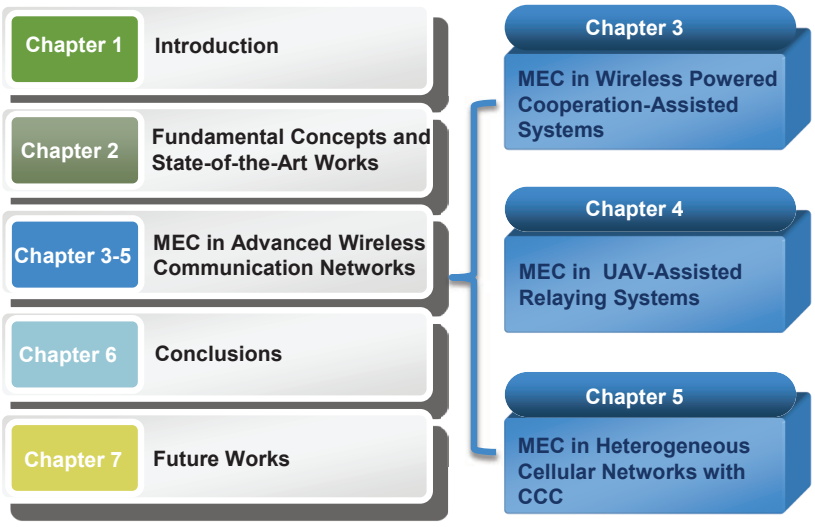


Figure 1.1: The figure of the thesis organization.

works.

Chapter 3: Mobile Edge Computing in Wireless Powered Cooperation-Assisted Systems. A wireless powered cooperation-assisted MEC architecture based on a WPCN is studied in this chapter, in which two near-far UEs are energized by the AP through WPT. Partial computation offloading is utilized to offload part or all of UEs' computation tasks to the MEC server co-located at the AP. A harvest-then-offload protocol with a block-based time division mechanism is proposed, where the technology of cooperative communications is leveraged to overcome the double-near-far effect in WPCNs. A low-complexity algorithm is proposed to effectively solve the AP's transmit energy minimization (APTEM) problem. The numerical results not only verify that the proposed cooperative computation offloading scheme can achieve a significant performance improvement but also demonstrate the effectiveness of the scheme in handling computation-intensive latency-critical tasks and resisting the double-near-far effect in wireless powered MEC systems.

Chapter 4: Mobile Edge Computing in UAV-Assisted Relaying Systems. This chapter explores a UAV-assisted MEC architecture, where the computing resources at the UAV and the AP are cooperatively utilized to help the UEs complete their computation tasks through partial offloading. In addition, the energy-efficient LoS transmissions of the UAV have been fully exploited since the UAV not only serves as a mobile computing server to help the UEs compute their tasks but also as a relay to further offload UEs' tasks to the AP for computing. The weighted sum energy consumption (WSEC) of the UAV and the UEs is minimized under some practical constraints, and an alternating optimization algorithm is devised to properly solve the problem by addressing three subproblems iteratively. Numerical results are presented to show the optimized trajectories of the UAV under different scenarios and the significant performance enhancement by leveraging the proposed algorithm when compared with the existing benchmarks.

Chapter 5: Mobile Edge Computing in Heterogeneous Cellular Networks with Central Cloud Computing. In this chapter, we study the coexistence and synergy between the edge and central cloud computing in a heterogeneous cellular network (HetNet) with an MBS and multiple SBSs. The SBSs are empowered by edge clouds offering limited edge computing services for UEs, whereas the MBS provides high-performance central cloud computing services to UEs via restricted MIMO backhauls to their associated SBSs. An iterative algorithm based on decomposition is proposed to solve the problem of minimizing the system energy consumption while under the processing latency constraints at both the central and edge networks. Numerical results show that the proposed solution can achieve better performance than conventional schemes using edge or central cloud alone. Also, with large-scale antennas at the MBS, the unique features of massive MIMO

backhauls can significantly reduce the complexity of the proposed algorithm and obtain even better performance.

Chapter 6: Conclusions. This chapter summarizes the main conclusions of this thesis.

Chapter 7: Future Works. The future works based on this thesis are discussed in this chapter. We first discuss some straightforward methods to extend the work in Chapter 3 to more general settings. Then, we propose a wireless powered MEC architecture with a cooperative UAV, which can be regarded as an extension of the work in Chapter 4 by introducing the technologies of WPT and time allocation, in order to further enhance the sustainability and flexibility of the UAV-assisted MEC systems. Last, a cache-enabled multi-cell MEC scenario is demonstrated, which is promising to address the resource allocation problems related to both edge computing and caching.

1.4 List of Publications

1.4.1 Journal Papers

[J1] **Xiaoyan Hu**, Kai-Kit Wong, and Kun Yang, "Wireless Powered Cooperation-Assisted Mobile Edge Computing," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2375 - 2388, Apr. 2018. (reference [70])

[J2] **Xiaoyan Hu**, Kai-Kit Wong, Kun Yang, and Zhongbin Zheng, "UAV-Assisted Relaying and Edge Computing: Scheduling and Trajectory Optimization," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4738 - 4752, Oct. 2019. (reference [71])

[J3] **Xiaoyan Hu**, Lifeng Wang, Kai-Kit Wong, Meixia Tao, Yangyang Zhang, and Zhongbin Zheng, "Edge and Central Cloud Computing: A Perfect Pairing for High Energy Efficiency and Low-Latency," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1070 - 1083, Feb. 2020. (reference [72])

[J4] **Xiaoyan Hu**, Kai-Kit Wong, and Yangyang Zhang., "Wireless Powered Edge Computing with Cooperative UAV: Task, Time Scheduling and Trajectory Design," **accepted in** *IEEE Transactions on Wireless Communications*. (reference [73])

1.4.2 Conference Papers

[C1] **Xiaoyan Hu**, Kai-Kit Wong, and Kun Yang, "Power Minimization for Cooperative Wireless Powered Mobile Edge Computing Systems," *2018 IEEE International Conference on Communications (ICC)*, pp. 1-6, Kansas City, MO, USA, May, 2018. (reference [74])

[C2] **Xiaoyan Hu**, Kai-Kit Wong, Kun Yang, and Zhongbin Zheng, "Task and Bandwidth Allocation for UAV-Assisted Mobile Edge Computing with Trajectory Design," *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-6, Waikoloa, HI, USA, Dec. 2019. (reference [75])

[C3] **Xiaoyan Hu**, Lifeng Wang, Kai-Kit Wong, Meixia Tao, Yangyang Zhang, Zhongbin Zheng, "The Synergy of Edge and Central Cloud Computing with Wireless MIMO Backhaul," *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-6, Waikoloa, HI, USA, Dec. 2019. (reference [76])

[C4] **Xiaoyan Hu**, Kai-Kit Wong, and Zhongbin Zheng, "Wireless Powered

Mobile Edge Computing with Cooperated UAV," 20th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1-5, Cannes, France, July, 2019. (reference [77])

Chapter 2

Fundamental Concepts and State-of-the-Art Works

2.1 Mobile Cloud Computing

Mobile devices such as smartphones, as the most effective and convenient communication tools, have become an essential part of our daily life. Due to the size-constrained and resource-limited property of mobile devices, they cannot effectively handle the computation-intensive or latency-critical tasks, and sometimes they are incapable to do so. To deal with the ever-increasing computation-intensive tasks generated by a large variety of mobile applications, the concept of CCC first emerges, which offloads these tasks to remote powerful data centers for computing, also known as central clouds. MCC is a refined concept, which integrates CCC into the mobile environment and facilitates mobile users to take full advantage of cloud resources [78, 79]. MCC can be defined as a combination of mobile networks and CCC [80, 81], and it has been considered as one of the most popular tools for mobile

users to access applications and services on the Internet.

Recent advances in virtualization and server interconnect architectures have boosted the use of datacenter infrastructures which is widely regarded as an enabling technology for services such as infrastructure as a service (IaaS), software as a service (SaaS), and platform as a service (PaaS). These kinds of services constitute the fundamental technologies behind cloud services. Based on these technologies, a lot of attractive advantages are endowed to CCC/MCC by allowing users to utilize infrastructures (e.g., servers, networks, and storages), softwares (e.g., application programs), and platforms (e.g., middleware services and operating systems) offered by cloud providers (e.g., Google, Amazon, and Salesforce) at low cost. In addition, CCC and MCC enable users to elastically utilize resources in an on-demand fashion.

Note that both CCC and MCC are in the vision with the centralization of computing, storage, and network management in the clouds, referring to data centers, backbone IP networks, and cellular core networks [4,5]. The basic function of CCC/MCC is computation offloading, i.e., shifting intensive computation from resource-limited UEs to powerful central cloud data centers. The cross-disciplinary nature of MCC has attracted significant attention from computer science and communications research communities in recent years, and extensive works on CCC/MCC have been conducted to explore the potential of central clouds. In order to prolong the battery lifetime of UEs and improve the computation performance, several system architectures using various code offloading frameworks, e.g., MAUI [7] and ThinkAir [8], were proposed. In [82], dynamic resource allocation using virtualization technology was studied to achieve overload avoidance and green computing by minimizing the number of physical machines. Also, a computation

offloading algorithm was proposed in [83] to deal with multiple services in workflow by leveraging MCC.

Although CCC/MCC can provide high-performance computing services for mobile users, it has one inherent drawback, i.e., the central clouds are usually located far away from users. Hence, accessing the CCC/MCC services induces excessive transmission latency, which will definitely increase the burden of the backhaul. Besides, it is easy to encounter the performance bottleneck considering the finite backhaul capacity and exponentially growing mobile data, which has led to the emergence of MEC in dealing with UEs' computation-intensive latency-critical tasks.

2.2 Mobile Edge Computing

The concept of MEC was firstly proposed by the ETSI in 2014, which was defined as a new platform that "provides IT and cloud computing capabilities within the RAN in close proximity to mobile subscribers" [17]. In other words, the rationale behind MEC is that the UEs' computation-intensive latency-critical tasks can be offloaded and completed at the edge of wireless networks by deploying edge cloud servers, i.e., the MEC servers, at the wireless APs, so as to liberate the resource-limited UEs from heavy computing workloads and prolong their battery lifetime. The MEC servers are typically small-scale data centers deployed by the cloud computing or telecom operators, which can be co-located with the wireless APs, e.g., the public Wi-Fi routers and BSs. In this way, the MEC allows the APs to have the ability of storage and processing, and thus guarantee that the UEs can be directly connected to the edge clouds. In comparison with the MCC, the MEC has four main advantages in the aspects of latency reduction, energy saving, context

awareness, and privacy/security enhancement, mainly due to the proximity to end users. The attractive advantages of MEC lead to the fact that it has been widely regarded as one of the key enablers to shape the future advanced wireless networks.

Similar to the MCC, MEC is also implemented based on a virtualization platform that leverages recent advancements in network functions virtualization (NFV), information-centric networks (ICNs), and software-defined networks (SDNs). Too be specific, NFV enables a single edge server to provide computing services to multiple UEs by creating multiple virtual machines (VMs) for simultaneously performing different tasks or operating different network functions of multiple users. The NFV-based MEC is promising to support the stringent requirements such as ultra-low latency and ultra-reliability of the forthcoming 5G services [84–87]. On the other hand, ICN provides an alternative end-to-end service recognition paradigm for MEC, shifting from a host-centric to information-centric architecture for implementing context-aware computing, such ac the computing tasks related to image or video processing [88, 89]. ICN-based MEC as well as MEC-enables caching are two interesting research directions for computing and caching resource allocation [90–92]. It should be noted that edge caching and computing are highly related for completing MEC tasks, and thus ICN plays an important role in MEC networks. Last, SDN allows MEC network administrators to manage services via function abstraction, achieving scalable and dynamic computing. Recently, the SDN-based MEC are exploited in LTE/LTE-A and vehicular ad hoc networks [93–95]. Actually, the functions of NFV, ICN and SDN are highly collaborated for enhancing the performance of data communication, computing, and caching. A main focus of MEC research is to develop these general network technologies so that they can be implemented at the network edges [96–98]. In a word, the off-the-

shelf technologies of cloud computing can be easily applied to MEC, which will definitely accelerate the development of MEC.

For practical deployment, several edge computing architectures have already been proposed, such as fog computing [99, 100], and also cloudlets [101]. Fog computing is a more flexible computing architecture consisting of highly heterogeneous fog computing nodes with different levels of computing ability such as routers and network gateways. Cloudlet is another concept of edge computing, where the computing resources are managed by cloudlet agents [101]. In wireless local area networks with Wi-Fi access, cloudlets run virtual machines to complete the computation tasks. Besides, multi-access edge computing (also using the same acronym "MEC" originated from mobile edge computing) has been introduced to support multiple access technologies including cellular, Wi-Fi, etc. [102]. Recently, MEC has been regarded as one of the key enablers to shape the future advanced wireless networks, which has attracted great attention from both the academia and the industry [14, 15, 103]. The standardization organizations and industry associations such as ETSI and 5G Automotive Association (5GAA) have identified a large number of use cases for MEC, from the intelligent video acceleration and application-aware performance optimization to V2X and massive machine-type communications (mMTC), etc. [18, 69, 104].

2.2.1 Computation Task Model

In order to properly conduct academic research related to MEC, we should first find a good way to model the computation tasks. Note that the computation tasks can be affected by various parameters such as task size, computation intensity, latency, bandwidth utilization, context awareness, scalability, and generality, etc.,

and thus developing accurate computation task models is highly sophisticated. It is known that energy consumption and latency, especially used for communication and computation, have been widely considered as two important performance metrics for MEC systems, and the objective is to complete the UEs' computation-intensive latency-critical tasks with high energy efficiency and low latency. Hence, to properly describe the properties related to energy consumption and latency, we adopt a reasonable and mathematically tractable computation task model in this thesis, which has also been widely used in the existing MEC literature.

For a given computation task with fixed computation task size, it can be fully characterized by a positive parameter tuple $[I, C, O]$. Here, I denotes the size (in bits) of the computation task-input data (e.g., the program codes and the input parameters), C is the amount of required computational resources for computing 1-bit of task-input data (i.e., the number of central processing unit (CPU) cycles required), also known as the computation workload/intensity, $O \in (0, 1)$ is the ratio of task-output data size to that of the task-input data, which means that computing I bits of task-input data will generate OI bits of task-output data for the specific UE. In addition, the parameters in the task tuple of $[I, C, O]$ can be obtained through task profilers by applying the methods, e.g., call graph analysis [7, 14, 105–107]. Note that this computation task model tuple not only captures the essential properties of mobile applications related to the computation and communication demands but also enables mathematical tractability shown in the following Sections 2.2.3 and 2.2.4. Besides, this model allows rich task modeling flexibility in practice and can be easily extended to consider other kinds of resources by introducing more parameters into the tuple. For example, the latency-critical computation tasks usually have a latency constraint, and thus a parameter T could be added into the

tuple to indicate the maximum tolerable latency or deadline for the computation task.

In terms of the sizes of the computation results (task-output data), the computation tasks can be generally divided into two groups as follows:

- **Computation Tasks with Negligible Computing Results:** For some computation tasks, the sizes of the task-output data, i.e., OI, are much smaller than the sizes of the task-input data I, like several orders of magnitude lower than I. For instance, the computation task-output data may be just a few command or control bits for some applications related to surveillance or system control, while the corresponding computation task-input data usually measured by Kbit or Mbit. In this case, the parameter O is usually with a very small value. Hence, the downloading overheads such as time and energy consumption for delivering the task-output data from the remote MEC servers back to the corresponding UEs are negligible and usually can be ignored.

- **Computation Tasks with Non-Negligible Computing Results:** In contract, for some computation tasks with a larger parameter O, the sizes of the task-output data OI are comparable to those of the task-input data I. For example, the tasks of video compression, even though the sizes of the compressed videos are much less than but still comparable to the input data sizes. In this case, the downloading overheads of time and energy consumption for delivering the computation task-output data from the remote MEC servers back to the corresponding UEs should be taken into consideration.

2.2.2 Computation Offloading Modes

According to the structural characteristics of various applications or computation tasks, different computation offloading modes should be leveraged to deal with different computation tasks. In this subsection, we introduce two computation offloading modes used in this thesis, respectively corresponding to the partial offloading mode and binary offloading mode, which are also popularly used in existing state-of-the-art literature on MCC and MEC.

- **Partial Offloading Mode:** Many mobile applications are composed of multiple procedures or components, making it possible to implement fine-grained (partial) computation offloading. Specifically, the computation task-input data are bit-wise independent and can be arbitrarily divided to facilitate parallel trade-offs between local computing at the UEs and computation offloading to other MEC servers with stronger computing capabilities. For the partial offloading tasks, the partition of the task-input data for parallel computation, i.e., task allocation, is necessary and has a great effect on the system performance.

- **Binary Offloading Mode:** For some atomic highly integrated computation tasks or relatively simple tasks, they cannot be partitioned and have to be completed as a whole either locally at the UEs or offload to the remote MEC servers. For the binary offloading tasks, mode selection (local computing mode or computation offloading mode) plays an important role and needs to be properly addressed.

2.2.3 Communications in MEC Systems

In MEC systems, communications act as an essential part for completing users' computation tasks, which typically happen between UEs and APs (with co-located MEC servers) through wireless channels. For computation tasks with negligible task-output data, communications mainly correspond to the computation offloading from UEs to the MEC servers, while for computation tasks with non-negligible task-output data, communications are also necessary for downloading the computation results from the MEC servers to UEs. In fact, the wireless APs not only provide wireless interfaces for the MEC servers but also enable the access to the remote central clouds (large-scale data centers) through backhaul links, thus assisting the MEC servers to further offload some computation-intensive tasks to enjoy the more powerful computing capabilities at the central clouds. In addition, for the mobile devices that cannot communicate with the APs directly due to insufficient wireless interfaces or severe blockage, device-to-device (D2D) communications through neighboring devices provide the opportunity to forward the computation tasks to MEC servers. Furthermore, D2D communications also enable the peer-to-peer cooperation on resource sharing and computation-load balancing within the clusters of mobile devices.

Next, we will analyze the communications in MEC systems from the two widely used performance metrics, i.e., latency and energy consumption. According to the Shannon-Hartley theorem [108, 109], the maximum achievable communication rate (in bits per second), i.e., the channel capacity, of a wireless additive white Gaussian noise (AWGN) channel can be expressed as

$$R = B \log_2 \left(1 + \frac{S}{N_0}\right), \tag{2.1}$$

where B is the bandwidth of the wireless channel in hertz (Hz); S indicates the average received signal power over the bandwidth, measured in watts (W); N_0 denotes the average power of the noise and interference over the bandwidth, measured in W; and $\frac{S}{N_0}$ is the signal-to-interference-plus-noise ratio (SINR) at the receiver. Normally, the average received signal power S can be further expressed as $S = Ph$, where P and h denote the transmit power and the effective channel gain, respectively. In other words, the wireless communication rate of a UE/AP is positively correlated to the transmit power and the effective channel gain of the corresponding UE/AP. It should be noted that the channel capacity can be achieved by employing a capacity-approaching code when large block lengths or computational tasks are considered. A more general model for the achievable rate can be expressed as $R = B\log_2\left(1 + \frac{S}{\Gamma N_0}\right)$, where Γ represent the gap between the channel capacity and the a specific modulation and coding scheme, and $\Gamma = 1$ when a capacity-approaching code is employed.

Based on the computation task model mentioned in Section 2.2.1, i.e., $[I, C, O]$, the communication latency for offloading I bits of computation task-input data from a UE to the MEC server can be calculated as

$$t_{\text{off}} = I/R_{\text{off}}, \tag{2.2}$$

where R_{off} is the corresponding communication rate for computation offloading based on (2.1). Accordingly, the energy consumption used for offloading the I bits of task-input data to the MEC server is given as

$$E_{\text{off}} = P_{\text{off}} t_{\text{off}} = P_{\text{off}} I/R_{\text{off}}, \tag{2.3}$$

where P_{off} is the UE's transmit power for computation offloading. As we described above, R_{off} is monotonically increasing versus P_{off}, and thus it is easy to note that there exists a performance tradeoff between the communication latency and energy consumption by adjusting UE's transmit power P_{off}. To be specific, the communication latency can be reduced by increasing P_{off} but at the cost of increasing the energy consumption used for communications, and vice versa. Hence, the UEs' transmit power for computation offloading is an important parameter for resource allocation in MEC systems which should be properly adjusted so as to achieve a good balance between the communication latency and energy consumption.

2.2.4 Computation in MEC systems

Computation also plays an important role in MEC systems for completing the UEs' computation tasks. Similarly, in this part, we mainly pay attention to the analysis of the energy consumption and latency related to computation in MEC systems.

The energy consumption of a computing server/processor is jointly determined by the usage of the CPU, storage, memory, and network interfaces, etc. Since the CPU contribution is dominant among these factors, it is the main focus widely used in the existing related literature. As for the CPU power, it consists of the dynamic power, the short circuit power, and leakage power, in which the dynamic power dominates and the other components are negligible compared with the dynamic power [14]. As a result, we only take the dynamic power into account, denoted as P_{comp}, which is proportional to the product of $V^2 f$ under the assumption of a low CPU voltage, where V and f are the corresponding circuit supplied voltage in volt (V) and the CPU clock frequency in cycles/second, respectively [14,30]. It is further noticed in [110–112] that, the clock frequency of the computing server/processor's

CPU chips, i.e., f, is approximately linearly proportional to the voltage supply V. In other words, P_{comp} should be linearly proportional to f^3, and thus can be written as $P_{\text{comp}} = \kappa f^3$, where κ is the effective capacitance coefficient that depends on the chip architecture of the computing server/processor. Hence, the unit energy consumption of the computing server/processor for operating each CPU cycle can be denoted as

$$E_{\text{unit}} = P_{\text{comp}} t_{\text{comp}} = \kappa f^3 * (1/f) = \kappa f^2, \tag{2.4}$$

where $t_{\text{comp}} = 1/f$ is the time duration for one CPU cycle [14]. Based on the computation task model mentioned in Section 2.2.1, i.e., $[I, C, O]$ with I bits of task-input data and each bit requiring C CPU cycles for computing, the energy consumption of computation for completing this task can be calculated as

$$E_{\text{comp}} = ICE_{\text{unit}} = \kappa IC f^2. \tag{2.5}$$

Accordingly, the computation latency for completing the task $[I, C, O]$ by operating IC CPU cycles can be expressed as

$$t_{\text{comp}} = IC/f. \tag{2.6}$$

To efficiently use the energy for computation, the computing servers/processors can leverage the dynamic voltage and frequency scaling (DVFS) technique. In this way, the energy consumed for computation can be adaptively controlled by adjusting their CPU frequency for each CPU cycle [20]. Denoting the adjustable CPU frequency for the i-th CPU cycle as f_i, then the energy consumption

of computation for completing the task $[I, C, O,]$ can be calculated as

$$E_{\text{comp}}^{\text{DVFS}} = \kappa \sum_{i=1}^{IC} f_i^2, \tag{2.7}$$

and the corresponding computation latency is described as

$$t_{\text{comp}}^{\text{DVFS}} = \sum_{i=1}^{IC} 1/f_i. \tag{2.8}$$

Another kind of DVFS computation is that the CPU frequency is fixed during a given slot and adaptively changes among different slots. In this case, we respectively denote the n-th slot length and the corresponding CPU frequency during this slot as τ_n and f_n, for $n = 1, 2, \cdots, N$, where N is the total number of slots. Hence, in order to complete the computation task $[I, C, O]$, the following equation should be satisfied

$$I = \sum_{n=1}^{N} I_n = \sum_{n=1}^{N} \tau_n f_n / C, \tag{2.9}$$

where $I_n = \tau_n f_n / C$ is the completed task-input bits during the slot n. Accordingly, the total computation energy consumption and latency for completing the computation task $[I, C, O]$ can be respectively calculated as

$$E_{\text{comp}}^{\text{hyb}} = \kappa \sum_{n=1}^{N} \tau_n f_n^3 = \kappa \sum_{n=1}^{N} I_n C f_n^2, \tag{2.10}$$

$$t_{\text{comp}}^{\text{hyb}} = \sum_{n=1}^{N} \tau_n = \sum_{n=1}^{N} I_n C / f_n. \tag{2.11}$$

From the above analysis, we can observe that there also exists a performance trade-off between the computation energy consumption and latency through ad-

justing the computing server/processor's CPU clock frequency f. Specifically, increasing f will definitely reduce the computation latency but at the cost of increasing the energy consumption used for computing, which is vice versa. This trade-off indicates that the computing server/processor's CPU clock frequency f also plays a significant role in resource allocation in MEC systems, which should be properly controlled in order to achieve a good balance between the computation energy consumption and latency.

2.2.5 Joint Design of Computation and Communication/Radio Resource Management

The broadcast nature and random variations of wireless channels in time, frequency, and space make it important to seamlessly integrate the control of computation and communication/radio resource management, and it is also crucial for designing high energy-efficient and low-latency MEC systems. For instance, when the wireless channels are in deep fading, the reduction in execution latency by remotely completing the computation tasks through computation offloading may not be sufficient to compensate for the increase of communication latency due to the steep drop in transmission-data rates. It is true that increasing transmit power for offloading can increase the data rate, but also lead to higher communication energy consumption. For such cases, it is desirable to defer the computation offloading until the channel gains are favorable or switch to alternative frequency/spatial channels with better quality for offloading. The above considerations necessitate the joint design of resource management for computation offloading and wireless communications, which should be adaptive to the time-varying channels based on the channel state information (CSI). For the deployment of wireless technologies in MEC systems,

the communication and networking protocols need to be redesigned to integrate both the computing and communication infrastructures, so as to effectively improve the computation efficiency.

2.2.6 State-of-the-art MEC Works

The cross-disciplinary nature of MEC plays an important role of joint computational and radio resource management in achieving energy-efficient or delay-optimal MEC performance. Recent years have witnessed the encouraging progress on this topic for both single-user [20–27] as well as multiuser [28–36] MEC systems.

For single-user MEC systems, an energy-optimal edge computing architecture under a stochastic wireless channel was considered in [20], where the optimal offloading decision policy by comparing the energy consumption of optimized local computing (with variable CPU cycles) and offloading (with variable transmission rates) was given. Later in [21], a dynamic offloading scheme with adaptive long term evolution (LTE)/Wi-Fi link selection was proposed to improve the energy efficiency. Another dynamic offloading scheme with energy harvesting was addressed in [22] to reduce the execution cost, including the execution latency and task failure, by leveraging the Lyapunov optimization technique. The tradeoff between energy consumption and latency in information transmission and computation was analyzed in [26], where a UE offloaded its application tasks to an SBS for processing. The energy-delay tradeoff in single-user MEC systems with a multicore UE and heterogeneous types of mobile applications were investigated in [23] and [24], respectively. In [25], a Markov decision process approach was adopted to handle a delay minimization problem, where the computation tasks were scheduled based on the queueing state of the task buffer, the execution state of the local

processing unit, as well as the state of the transmission unit. Later in [27], the scenario of a UE with multiple tasks was considered, where multiple APs assisted the UE to reduce its total task execution latency and energy consumption.

As for the multiuser MEC systems, joint radio-and-computational resource management becomes more complicated. An initial investigation for multi-user MEC systems with delay-tolerant applications was conducted in [28], which, however, only focused on computational resource scheduling and failed to address radio resource management. A multi-cell MEC offloading system was considered in [29], where the radio and computation resources were jointly allocated to minimize the overall energy consumption of users under offloading latency constraints. In [30], the distributed offloading decision-making problem was formulated as a multiuser computation offloading game to explore both energy-and-latency minimizations at mobile users. Optimal energy-efficient resource allocation for multiple users was addressed in [31] based on time-division multiple-access (TDMA) and orthogonal frequency-division multiple-access (OFDMA) systems. The cooperation among clouds was investigated in [32] to maximize the revenues of clouds and meet the demands of UEs via the resource pool sharing. In [33], stochastic resource management of multiple users resorting Lyapunov optimization was considered with the objective of minimizing the long-term average weighted sum power consumption of the UEs and the MEC server, subject to a task buffer stability constraint. Later in [34], an energy-aware offloading scheme was proposed to tradeoff between users' energy consumption and the execution latency for computation offloading. The sum of computation efficiency defined as the calculated data bits divided by the energy consumption was maximized in [35] with iterative and gradient descent methods. A multi-cell and multi-server MEC system was considered in [36], where joint task

offloading and resource allocation was addressed to maximize the task offloading gain.

In addition, the technology of MEC also plays an important role in promoting the development of IoT. It is known that IoT devices may lack computing capability, while MEC is capable to achieve edge execution which avoids frequent delivery of massive computing tasks to the core networks with central cloud for computing, and thus MEC can help IoT devices reduce the computing latency and backhaul congestion [10, 37]. The survey work [10] presented a comprehensive overview of fog computing in IoT networks and illustrated how fog computing tackles the challenges in IoT networks. In [37], Lyapunov optimization techniques were adopted to develop an online MEC scheduling solution with partial knowledge of the IoT network.

Recent works related to edge computing also focus on multi-service scenarios. For example, [38] considered a single MEC server with storage capability and attempted to maximize the revenue of providing both the computing and caching services. In [39], a D2D fogging was explored to achieve energy-efficient task completion by sharing computation and communication resources amongst mobile devices. A blockchain-based platform was also considered for video streaming with MEC in [40], and an incentive mechanism was proposed to facilitate the cooperation of different nodes. Most recently, user cooperation was also adopted as an effective method to improve the MEC performance [41], where a three-node MEC system was considered to exploit joint computation and communication cooperation for reducing the total energy consumption of the system.

The complementary benefits between the edge and central cloud have driven research towards the coexistence and cooperation between the edge and central

clouds [113]. One such example was [114] where delay-aware scheduling between local and Internet clouds was studied, and a priority-based cooperation policy was given to maximize the total successful offloading probability. The placement and provisioning of virtualized network functions were explored in [115], in which a QoS-aware optimization strategy was proposed over an edge-central carrier cloud infrastructure. Also, the work in [116] considered that an edge server and a central cloud coexist to complete the UEs' computations cooperatively, where a wired connection was assumed between the edge and the central cloud. However, the existing works [114–116] considering the coexistence of edge and central cloud computing either focus on delay-aware priority scheduling, virtualized resource allocation, or offloading with wired backhaul. The issues related to offloading decision and resource allocation of hybrid edge/central cloud computing networks with wireless backhaul have not been thoroughly studied, especially from the viewpoint of communications [14]. Therefore, we completed the works [J3] and [C3] ([72] and [76]), where the deployment of heterogeneous edge and central clouds was studied to leverage the easy access of edge clouds and the abundant computing resources at the central cloud, mainly from the viewpoint of communications by considering cloud selection, resource allocation, and the physical properties of the wireless backhauls.

2.3 Wireless Power Transfer

Even though MEC has many advantages as we mentioned in the previous section, taking the full benefits of powerful computational resources at the edges still faces several challenges. Among them, the insufficient power supply is one major limitation of conventional battery-based UEs. The computing performance may

be compromised due to the lack of energy supply, i.e., mobile applications will be terminated and UEs will be out of services if their batteries are running out. It is true that this issue can be addressed to some degree by using larger batteries or recharging the batteries regularly. However, using larger batteries at the UEs implies increased hardware cost, which is not desirable. On the other hand, recharging batteries frequently is reported as one of the most unfavorable characteristics of UEs, and it may even be impossible in certain application scenarios, e.g., for sensors embedded in building structures or wearable devices inside human bodies. It therefore makes sense to leverage the technology of WPT, which is known as a promising solution to provide convenient and sustainable energy supplies to wireless networks. The WPT utilizes the radio frequency (RF) wave as the carrier of energy to wirelessly charge UEs, so that user devices are not power-limited by their batteries but can be energized remotely, e.g., [42–55]. WPT, particularly in the form of simultaneous wireless information and power transfer (SWIPT) [44–46] and WPCNs [47–55] have recently been considered as two important paradigms to provide genuine sustainability for mobile communications.

2.3.1 Energy Harvested from WPT

The mobile devices, wearable devices, unmanned aerial devices, and sensors, etc., can all be treated as UEs that are able to harvest energy from the APs or dedicated power beacons that broadcast RF energy through WPT. Assume that the energy transmit power of the AP is denoted as P_0, and the effective channel gain between the AP and the specific UE is h, and thus the harvested energy for this UE during a

time slot T can be calculated as

$$E^{\mathrm{harv}} = \nu P_0 h T, \tag{2.12}$$

where the linear energy harvesting model is adopted since we assume that the input RF power of UEs are within the linear regime of the rectifier, and ν is the energy conversion efficiency of the UE. Note that the energy transmission efficiency can be highly improved by leveraging some advanced communication techniques to improve the effective channel gain h, such as using the technology of energy beamforming if the AP is equipped with multiple antennas.

The WPT technique can support the UEs with sustainable energy supply, and the extra energy can be stored by the UEs for their future operations. For these UEs, an energy harvesting causality constraint should be satisfied in each time slot, i.e.,

$$E^{\mathrm{cons}} \leq E^{\mathrm{harv}} + E^{\mathrm{sav}}, \tag{2.13}$$

where E^{cons} is the UE's energy consumption during the corresponding slot, and E^{sav} is the UE's energy savings from the previous time slots.

2.3.2 MEC Works in Networks with WPT

The wireless powered MEC systems are typically WPCNs, where the RF energy transmissions are from APs to UEs through downlink channels while the information transmissions for computation offloading are from UEs to APs through the uplink channels. As we mentioned before, the combination of MEC and WPT is a promising solution to release the burden of resource-limited UEs. Many recent works have witnessed the possible synergy integrating MEC with WPT

[52–55]. An interesting work in [52] considered a wireless powered single-user MEC system, where a single-antenna sensor harvested RF energy from a dedicated BS for computation offloading, in which binary offloading was investigated to maximize the computing probability. More recently in [53], an energy-efficient wireless powered multiuser MEC system combining with a multi-antenna AP was considered. The optimal transmit energy beamforming of the AP, the offloading decision, and the resource allocation for minimizing the energy consumption at the AP were obtained. Unlike the considered network in [53] where wireless power transfer and computation offloading were operated over orthogonal frequency bands, the work [54] designed a new time frame that the AP first broadcast the RF energy to the UEs and then the energy-constrained UEs offloaded their tasks to the AP at their allocated time slots, where the computation rate was maximized with the binary offloading mode. In [55], BSs were powered by hybrid energy supplies including green energy and grid power, and a green-energy aware cloudlet solution was proposed to minimize the total grid power consumption.

2.3.3 Double-Near-Far Effect in WPCNs

As we mentioned before, the wireless powered MEC systems are typically WPCNs. However, WPCNs are susceptible to suffer from the so-called "double-near-far" effect, which occurs because the farther UEs from an AP harvest less energy and are also required to communicate in longer distances [47–49]. In other words, if two identical UEs are powered by an AP and have equally-intensive computational tasks to be offloaded for computing at the MEC server located at the AP, the farther device harvesting less energy will consume more energy for computation offloading due to the doubled distance-dependent signal attenuation over both the downlink energy

harvesting and uplink computation offloading. It is known that user cooperation is an effective way to improve the capacity, coverage, and diversity performance in conventional wireless communication systems. Recent works [49–51, 117, 118] show that cooperation among near-far users in WPCNs is also capable to resist the double-near-far effect in WPCNs, so as to improve performance of WPCNs.

Based on the analysis above, we understand that user cooperation should be an effective way to deal with the double-near-far effect in wireless powered MEC networks which are typically WPCNs. It is against this background that we completed the works [J1] and [C1] ([70] and [74]), which introduce the technology of user cooperation into a three-node wireless powered MEC network. In this work, two UEs are powered by the AP through WPT and the nearer UE to the AP is selected to act as a relay to help offload the farther UE's computation tasks so as to satisfy the latency constraint of tasks as well as reduce the total energy consumption of the AP. It is demonstrated that the user cooperation is of great value in resisting the double-near-far effect in wireless powered MEC networks.

2.4 UAV-Enabled Communications

UAVs, also commonly known as drones, are aircrafts piloted by remote control or embedded computer programs without human onboard. Recently, the cellular-based UAV-enabled wireless communications have drawn great attention from both academia and industry due to the attractive advantages of the UAVs for their easy deployment, flexible movement, and LoS connections, etc [58]. Thanks to the almost ubiquitous coverage of the cellular network worldwide as well as its advanced communication technologies, it is capable to support the UAV-ground communications in a cost-efficient manner, which significantly promotes

the development of cellular-based UAV-enabled communications. The forthcoming 5G cellular network is expected to achieve the peak data rate of 10 Gbits/second with only 1 millisecond round-trip latency, which in principle is adequate for high-rate and delay-sensitive UAV communication applications such as real-time video streaming and data relaying. In this way, the requirements for UAV-enabled communications for both the control and payload communications can be potentially met, regardless of the density of UAVs as well as their distances with the corresponding ground nodes.

Generally, the cellular-based UAV communications can be partitioned into two categories, i.e., cellular-connected UAV communications and UAV-assisted communications [56–58]. The UAVs in cellular-connected UAV communications are considered as aerial users that access the cellular networks from the sky for wireless communications. Cellular-connected UAV communication is a cost-effective way for wireless communications since it reuses the millions of cellular BSs worldwide without the need of building new infrastructures dedicated for unmanned aerial systems (UAS) only. In this way, the cellular-connected UAV communication is expected to be a win-win technology for both UAV and cellular industries, with rich business opportunities to explore in the future. In contrast, the UAVs in UAV-assisted communications are normally regarded as aerial communication platforms such as APs, BSs, and relays, to assist the terrestrial wireless communications by providing access interfaces from the sky. UAVs as aerial APs can bring many attractive advantages compared to conventional terrestrial communications with typically static APs. First, UAV-mounted APs can be swiftly deployed on demand, which is especially appealing for application scenarios such as temporary or unexpected events, emergency response, search, and rescue, etc. Besides, UAVs

as aerial APs are more likely to have LoS connections with their ground users thanks to their high altitude above the ground, thus providing more reliable links for communications as well as multiuser scheduling and resource allocation. In addition, an additional degree of freedom can be achieved from the controllable altitudes of the UAVs, which makes it possible to enhance the communication performance by dynamically adjusting their locations in three-dimensional (3D) to cater for the terrestrial communication demands.

Based on the above advantages of UAV-enabled communications, it is of great benefits to introduce UAV-enabled communications into MEC networks. It is true that MEC has been widely regarded as a key technology for enhancing the computational capabilities of small devices by allowing them to offload the computation-intensive tasks to nearby MEC servers (e.g., APs). However, for users located at the cell edge, such an offloading strategy may even cause more transmission energy and/or longer delay than local computation due to the limited communication rate with the AP. To address this problem, UAVs with highly controllable mobility can be used as the flying cloudlets/servers to achieve more efficient computation offloading for the users by moving significantly closer to them. Hence, it is a great attempt to leverage the technology of the UAV in MEC systems, and the performance improvement of the UAV-enabled MEC architectures has been shown to be substantial in literature [65–68].

2.4.1 UAVs' Propulsion Energy Consumption

For UAV-enabled MEC networks, the UAVs' energy consumption should include that utilized for task transmissions (offloading or downloading), task computation, and propulsion, where the additional propulsion energy is used to remain the

UAVs aloft and moving freely over the air. Hence, the energy-efficient design of UAV-enabled MEC networks is more involved than that for the conventional terrestrial MEC systems which consider the transmission and computation energy only. Note that the energy consumed for task transmissions and computation can refer to the corresponding expressions given in subsections 2.2.3 and 2.2.4, respectively. However, the propulsion energy highly depends on the types of UAVs. In practice, there are many types of UAVs that are applicable for numerous and diversified applications. In terms of wing configuration, fixed-wing and rotary-wing UAVs are the two main types of UAVs that have been widely utilized in existing works. Typically, fixed-wing UAVs have higher maximum flying speed and can carry greater payloads for traveling longer distances as compared to rotary-wing UAVs, while their disadvantages lie in that a runway or launcher is needed for take off/landing as well as that hovering at a fixed position is impossible. In contrast, rotary-wing UAVs are able to takeoff/land vertically and remain static at a hovering location. The propulsion energy consumption for these two kinds of UAVs are quite different, which is described as follows [58].

- **Fixed-wing UAV propulsion energy consumption model:** For a fixed-wing UAV in straight-and-level flight with constant speed v meter/second (m/s) in the duration τ, the propulsion energy consumption consists of the parasite and induced energy, which can be expressed in a closed form as

$$E_{\mathrm{U,prop}}^{\mathrm{fixed}} = \tau\Bigg(\underbrace{\theta_1 v^3}_{\text{parasite}} + \underbrace{\frac{\theta_2}{v}}_{\text{induced}} \Bigg), \tag{2.14}$$

  where θ_1 and θ_2 are two parameters related to the UAV's weight, wing area, wing span efficiency, and air density, etc.

- **Rotary-wing UAV propulsion energy consumption model:** In contrast, for a rotary-wing UAV in straight-and-level flight with speed v in the duration τ, the propulsion energy consumption consists of the parasite, induced, and the blade profile energy, which is expressed as

$$E_{\mathrm{U,prop}}^{\mathrm{rotary}} = \tau \Bigg[\underbrace{p_0 \left(1 + \frac{3v^2}{U_{\mathrm{tip}}^2}\right)}_{\text{blade profile}} + \underbrace{p_i \left(\sqrt{1 + \frac{v^4}{4v_0^4}} - \frac{v^2}{2v_0^2}\right)^{\frac{1}{2}}}_{\text{induced}} + \underbrace{\frac{1}{2} f_0 \varrho s A v^3}_{\text{parasite}} \Bigg], \tag{2.15}$$

  where p_0 and p_i respectively denote the blade profile power and induced power in hovering status that depend on the aircraft weight, air density ϱ, rotor disc area A, etc., U_{tip} represents the tip speed of the rotor blade, v_0 is known as the mean rotor induced velocity in hovering, f_0 and s are the fuselage drag ratio and rotor solidity, respectively.

For both types of UAVs, the energy consumption for propulsion consists of at least two components: the parasite energy and the induced energy. The parasite energy is needed to overcome the parasite drag caused by the moving of the aircraft in the air, while the induced energy is used for overcoming the induced drag resulted from the lift force to maintain the aircraft airborne. Besides, for both two kinds of UAVs, the parasite power increases in cubic with the aircraft speed v, while the induced power decreases as v increases, with a more complicated expression for rotary-wing UAVs than fixed-wing UAVs. Compared to the fixed-wing UAVs, the rotary-wing UAVs has one additional propulsion energy term: the blade profile energy, which is needed to overcome the profile drag due to the rotation of blades. From the two expressions in (2.14) and (2.15), we can observe that the required energy consumption of fixed-wing UAV is infinity for the extreme case with $v =$

0, whereas that of rotary-wing UAVs is given by a finite value $\tau(p_0 + p_i)$. This corroborates the well-known facts that fixed-wing UAVs must maintain a minimum forward speed to remain airborne, while rotary-wing UAVs can hover with zero speed at fixed locations [58].

2.4.2 UAV-Related Works

Due to the attractive advantages of UAV for its easy deployment, flexible movement, and LoS connections, and so on, extensive UAV-enabled wireless communication works have been researched in recent years [56–64]. For instance, an energy-efficient UAV communication was investigated in [59], in which a UAV flew at a fixed altitude and had the initial and final locations preset on its trajectory design. In [60], the UAV-enabled mobile relaying systems were studied, where the throughput was maximized by optimizing the transmit power allocation and the UAV's trajectory. Recently, [61] proposed a generic framework for the analysis and optimization of the air-to-ground systems, and an optimum altitude for UAV in maximizing the coverage region with a guaranteed minimum outage performance was derived.

The technology of WPT was considered for UAV wireless networks in [62], and the UAV trajectory was optimized to maximize the sum energy or the minimum energy transferred to all the UEs. It was revealed that UAV-enabled WPT can significantly enhance the WPT performance over the traditional WPT systems where fixed energy transmitters are utilized. To better take the advantages of the dominated LoS air-ground links provided by UAVs, an more energy-efficient laser-beamed WPT technology has been utilized in recent wireless powered UAV-enabled architectures [119, 120], which is regarded as a viable solution to provide

unlimited endurance for UAVs in flight. Through providing a narrower energy laser beam, hundred of watts can be harvested at the laser power receiver [121], and the feasibility for laser-charged UAV has been verified by the field tests of LaserMotive company [122].

A UAV-based MEC system was investigated in [65], where a moving UAV equipped with a processing server was considered to help UEs compute their offloaded tasks. The total mobile energy consumption was minimized by jointly optimizing the task-bit allocation and the UAV trajectory using the successive convex approximation (SCA) method. Later in [66], a wireless powered UAV-enabled MEC system was studied, where the UAV was endowed with an energy transmitter and an MEC server to provide energy as well as MEC services for the UEs. The computation rate maximization problems were addressed under both the partial and binary computation offloading modes by alternating algorithms. A UAV-aided offloading scenario was considered at the edges of multiple cells in [67], in which the sum rate of edge users was maximized by optimizing the UAV's trajectory and user scheduling. In another study [68], the UAV acted as a UE rather than an MEC server, which was served by multiple cellular ground base stations to compute its offloaded tasks. The UAV's mission completion time was minimized by optimizing the resource allocation and the UAV trajectory through an SCA algorithm.

The aforementioned MEC works concentrate either on the cellular-based MEC networks where the UEs' tasks are completed by using the computing resources at the APs; or the UAV-enabled MEC architectures by exploiting the computing capabilities of the UAV processing servers. However, for the UEs with seriously degraded links to the AP due to severe blockage, it is impossible to take full use

of the computing resources at the AP directly. Besides, due to the size-constrained resource-limited property of the UAVs, it is risky to rely only on the UAVs to assist the UEs for completing their computation-intensive latency-critical tasks. For these reasons, we completed the works [J2] and [C2] ([71] and [75]), where a UAV-assisted MEC architecture was studied, and the computing resources at the UAV and the AP are utilized at the same time. In addition, the energy-efficient LoS transmissions of the UAV have been fully exploited since the UAV is not only served as a mobile computing server to help the UEs complete their computation tasks but also as a relay to further offload UEs' tasks to the AP for computing.

Chapter 3

Mobile Edge Computing in Wireless Powered Cooperation-Assisted Systems

This chapter is based on our works published in [J1] and [C1] ([70] and [74]).

3.1 Introduction

MEC has been widely regarded as a promising solution to liberate the resource-limited UEs from heavy computation workloads through helping them compute their offloaded computation-intensive latency-critical tasks. In order to further help the battery-based resource-limited UEs make full use of powerful computational resources at the edge servers, the technology of WPT is utilized to provide convenient and sustainable energy supplies for UEs. Besides, user cooperation is leveraged to effectively resist the double-near-far effect in WPCNs.

In this chapter, we study a wireless powered MEC system based on a WPCN, where two near-far UEs are energized by the AP through WPT. Partial computation offloading mode is adopted, so that the UEs can offload part or all of their computation-intensive latency-critical tasks to the AP connected with an MEC server or an edge cloud. A harvest-then-offload protocol is operated for UEs in an optimized time-division manner, so as to make better use of the system energy and time resources. Besides, to overcome the double-near-far effect on the farther UE in this WPCN, cooperative communications in the form of relaying via the nearer UE is considered for computation offloading of the farther UE. Our aim is to minimize the AP's total transmit energy through jointly optimize the AP's energy transmit power, UEs' offloading power, and time allocation, subject to the time allocation constraint, computation task constraints, and energy harvesting causality constraints. We first formulate the AP's transmit energy minimization (APTEM) problem and then prove that it can be equivalently transformed into a min-max problem, which can be optimally solved by a two-phase method. Numerical results demonstrate that the optimized wireless powered MEC system utilizing cooperation can achieve significant performance improvement in handling the UEs' computation-intensive latency-critical tasks and resisting the double-near-far effect caused by doubly path-loss in WPCNs.

3.2 System Model and Problem Formulation

3.2.1 System Model

We consider a wireless powered MEC system shown in Figure 3.1 that consists of a single-antenna AP (with an integrated MEC server), and two single-antenna

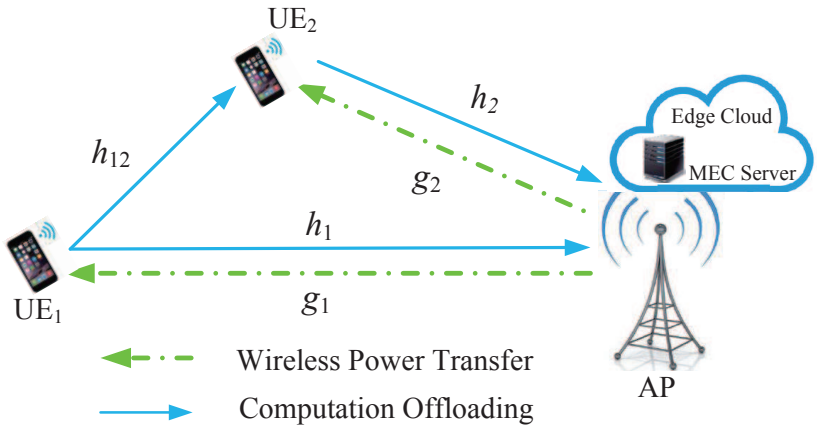


Figure 3.1: An illustration of the wireless powered cooperation-assisted MEC architecture, where the AP broadcasts RF energy to two near-far UEs through WPT and the UEs offload their computation tasks to the AP for computing by leveraging user cooperation.

UEs, denoted by UE_1 and UE_2, both operating in the same frequency band and each having a computation-intensive latency-critical task to be completed. A block-based TDMA structure is adopted where each block has a duration of T seconds. During each block, AP energizes the UEs in the downlink via WPT. Using the harvested energy, the two UEs accomplish their computation tasks in a partial offloading fashion [14], where the task-input bits are bit-wise independent and can be arbitrarily divided to facilitate parallel trade-offs between local computing at the UEs and computation offloading to the MEC server at the AP. After the AP computes the offloaded data, it sends the results back to the UEs. Note that local computing and downlink WPT can be performed simultaneously while wireless communications (for offloading) and WPT are non-overlapping in time considering half-duplex transmission for both two users. As a result, the harvest-then-transmit protocol proposed in [47] is employed in our model but for wireless powered computation offloading, which we refer to it as the harvest-then-offload protocol. Assuming that the AP has the perfect knowledge of all the channels and task-related parameters which can be obtained by feedback, the AP is designed to make

offloading decisions and allocate both radio and computational resources optimally so as to improve the system performance.

3.2.2 Computation Task Model

Each UE_k ($k \in \{1, 2\}$) has a computation-intensive and latency-critical task in each block, fully characterized by a positive parameter tuple $[I_k, C_k, O_k, T_k]$, where I_k denotes the size (in bits) of the computation task-input data (e.g., the program codes and input parameters), C_k is the amount of required computational resources for computing 1-bit of task-input data (i.e., the number of CPU cycles required), $O_k \in (0, 1)$ is the ratio of task-output data size to that of the task-input data, i.e., the output data size should be $O_k I_k$, and T_k in second (s) is the maximum tolerable latency. A UE can apply the methods (e.g., call graph analysis) in [7, 107] to obtain the information of I_k and C_k. Note that this model allows rich task modeling flexibility in practice and can be easily extended to consider other kinds of resources by introducing more parameters in the tuple. In this chapter, we assume that the maximum tolerable latency for two users is one block length, i.e., $T_1 = T_2 = T$.

3.2.3 User Cooperation Model for Computation Offloading

For computation-intensive latency-critical tasks with large input data sizes (large I_k) and strict latency constraints (small T), it would be difficult to rely upon local computing by UEs themselves to satisfy the latency constraint, and thus computation offloading may be necessary. Considering the double-near-far effect in our considered WPCN, cooperation amongst near-far UEs during offloading will help to improve the computation performance. Without loss of generality, it is assumed that UE_2 is nearer to the AP than UE_1, and we denote the distances

between AP and UE_1, AP and UE_2, UE_1 and UE_2 as d_1, d_2, and d_{12}, respectively, with $d_2 \leq d_1$. We also assume that $d_{12} \leq d_1$, and therefore it will be easier for UE_2 to decode the information sent by UE_1 than the AP, which makes such cooperative communications meaningful.

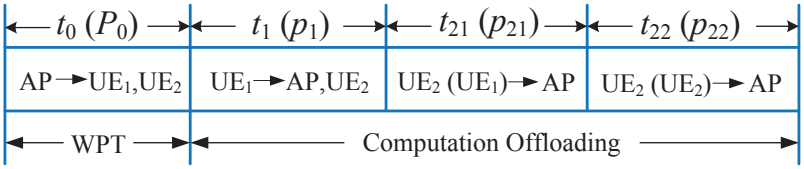


Figure 3.2: The time division structure for the harvest-then-offload protocol.

For an arbitrary single block, the time division structure is shown in Figure 3.2. During the first period t_0, AP broadcasts wireless energy to both UE_1 and UE_2 in the downlink with transmit power P_0. Assume that the two UEs have enough battery storages, and thus the energy harvested by each UE during the WPT period is given by

$$E_k = \nu_k g_k P_0 t_0, \; k \in \{1, 2\}, \tag{3.1}$$

where g_k is the downlink channel power gain from the AP to UE_k and $0 < \nu_k \leq 1$ is the energy conversion efficiency for UE_k. Note that no other sources of energy are available to carry out the computation tasks except from the WPT of the AP.

After the WPT period, UE_1 transmits its input-data-bearing information with average power p_1 from its harvested energy during the subsequent period t_1, and both the AP and UE_2 decode their respective received signals from UE_1. To overcome the double-near-far effect, during the remaining time of the block, the nearer user UE_2 will first relay the farther user UE_1's information with average power p_{21} over t_{21} amount of time and then transmits its own input-data-bearing

information to the AP with average power p_{22} over period t_{22}, all using its harvested energy. We denote the time allocation and power allocation vectors as $\mathbf{t} = [t_0, t_1, t_{21}, t_{22}]$ and $\mathbf{p} = [p_1, p_{21}, p_{22}]$, respectively. According to the results (Theorems 1–5) in [118], with a given pair of $\mathbf{t}$ and $\mathbf{p}$, the offloaded data size of UE_1 for remote computation at the AP should be the smaller value between the decoded data sizes at the AP and UE_2, i.e.,

$$L_1(\mathbf{t}, \mathbf{p}) = \min \left\{ L_{1,1}(\mathbf{t}, \mathbf{p}) + L_{1,2}(\mathbf{t}, \mathbf{p}), L_{1,12}(\mathbf{t}, \mathbf{p}) \right\}, \tag{3.2}$$

where $L_{1,1}(\mathbf{t}, \mathbf{p})$, $L_{1,2}(\mathbf{t}, \mathbf{p})$ and $L_{1,12}(\mathbf{t}, \mathbf{p})$ denote UE_1's offloaded data size from UE_1 to the AP, from UE_2 to the AP, and from UE_1 to UE_2, respectively, which are given by

$$L_{1,1}(\mathbf{t}, \mathbf{p}) = t_1 r_{1,1}(\mathbf{p}) = t_1 B \log_2 \left(1 + \frac{p_1 h_1}{N_0} \right), \tag{3.3}$$

$$L_{1,2}(\mathbf{t}, \mathbf{p}) = t_{21} r_{1,2}(\mathbf{p}) = t_{21} B \log_2 \left(1 + \frac{p_{21} h_2}{N_0} \right), \tag{3.4}$$

$$L_{1,12}(\mathbf{t}, \mathbf{p}) = t_1 r_{1,12}(\mathbf{p}) = t_1 B \log_2 \left(1 + \frac{p_1 h_{12}}{N_2} \right), \tag{3.5}$$

where $r_{1,1}(\mathbf{p})$, $r_{1,2}(\mathbf{p})$, and $r_{1,12}(\mathbf{p})$ are the transmission rates according to the channel achievable rates for offloading UE_1's input data. In the above expressions, h_1, h_2 are the uplink channel power gains from UE_1 and UE_2 to the AP, respectively, and h_{12} is the device-to-device channel power gain from UE_1 to UE_2.[1] Also, B is the channel bandwidth. N_0 and N_2 are respectively the receiver noise power at the AP and UE_2, and we further assume that $N_2 = N_0$ without loss of generality.

[1] All the channels mentioned in this chapter are quasi-static block fading channels. In order to investigate the effect of user cooperation in resisting the double-near-far effect caused by path loss, we mainly consider the case of $h_1 < h_{12}$, and thus $L_{1,1}(\mathbf{t}, \mathbf{p}) < L_{1,12}(\mathbf{t}, \mathbf{p})$ always holds.

Similarly, the offloaded data size of UE_2 for computing at the AP is described as

$$L_2(\mathbf{t}, \mathbf{p}) = t_{22} r_2(\mathbf{p}) = t_{22} B \log_2 \left(1 + \frac{p_{22} h_2}{N_0}\right), \tag{3.6}$$

where $r_2(\mathbf{p})$ denotes the transmission rate for offloading UE_2's input data. According to the task model, the offloaded data size of each user should not be greater than its corresponding input data size, i.e., $L_k(\mathbf{t}, \mathbf{p}) \leq I_k$, for $k \in \{1, 2\}$.

In practice, the MEC-integrated AP is capable of providing sufficient CPU computing capability, and thus the decoding and computation time spent at the AP can be ignores especially compared with those consumed by local computing at UEs themselves. It is assumed that the size of the computation task-out data, i.e., $O_k I_k$, is much smaller than that of the task-input data I_k in the considered application scenario of this chapter. For instance, the computation task-output data may be just a few command or control bits for some applications related to surveillance or system control, while the corresponding computation task-input data usually measured by Kbit or Mbit. In this case, the parameters $\{O_k\}_{k \in \mathcal{K}}$ are usually with very small values. Hence, the downloading overheads such as time and energy consumption for delivering the computation task-output data from the remote MEC server back to the corresponding UEs are negligible and usually can be ignored. For the nearer user UE_2, the decoding time for UE_1's information is also negligible compared with the wireless uplink transmission time for offloading both UE_1 and UE_2's extensive task-related information. For these reasons, we only consider the WPT time and the uplink offloading time as the total latency of the considered WPT-MEC system, and thus we obtain a latency constraint given by

$$t_0 + t_1 + t_{21} + t_{22} \leq T. \tag{3.7}$$

For each user, the energy required to receive its computed results from the AP is also considered negligible. Therefore, the energy consumption of UE_1 and UE_2 for computation offloading equals to the energy consumed for wireless transmissions, given by[2]

$$\begin{cases} E_{\text{off},1}(\mathbf{t},\mathbf{p}) = p_1 t_1, \\ E_{\text{off},2}(\mathbf{t},\mathbf{p}) = p_{21} t_{21} + p_{22} t_{22}. \end{cases} \tag{3.8}$$

3.2.4 Local Computing Model

Given a pair of time and power allocation vectors $(\mathbf{t},\mathbf{p})$, the offloaded data sizes $\{L_k(\mathbf{t},\mathbf{p})\}$ will be known, and hence the remaining input data of the corresponding computation tasks, i.e., $I_k - L_k(\mathbf{t},\mathbf{p})$, should be computed locally at UE_k, $k \in \{1,2\}$. For local computing, we assume that the CPU frequency is fixed as f_k for UE_k, which means that the two UEs are of limited computing resources. In order to satisfy the latency constraint, i.e., $(I_k - L_k(\mathbf{t},\mathbf{p}))\, C_k/f_k \leq T$, the offloaded data for UE_k should have a minimum size of $L_k(\mathbf{t},\mathbf{p}) \geq M_k^+$ with $M_k = I_k - f_k T/C_k$ where $a^+ = \max\{a, 0\}$. Under the assumption of a low CPU voltage that normally holds for low-power devices, the energy consumption per CPU cycle for local computing at UE_k can be denoted as $Q_k = \kappa_k f_k^2$, where κ_k is the effective capacitance coefficient that depends on the chip architecture. Hence, the energy consumption of UE_k for local computing can be expressed as

$$E_{\text{loc},k}(\mathbf{t},\mathbf{p}) = (I_k - L_k(\mathbf{t},\mathbf{p}))\, C_k Q_k,\ k \in \{1,2\}. \tag{3.9}$$

[2]All the energy consumption in this thesis uses the unit of Joule, abbreviated as J.

3.2.5 Problem Formulation

Based on the analysis above, we can obtain the energy saving for UE$_k$, $k \in \{1, 2\}$ as follows

$$E_{\mathrm{s},k}\left(P_0, \mathbf{t}, \mathbf{p}\right) = \nu_k g_k P_0 t_0 - E_{\mathrm{off},k}\left(\mathbf{t}, \mathbf{p}\right) - E_{\mathrm{loc},k}\left(\mathbf{t}, \mathbf{p}\right). \tag{3.10}$$

Furthermore, the APTEM problem for minimizing AP's transmit energy can be formulated as problem (P3.1) below

$$\begin{aligned}
(\mathrm{P3.1}): \min_{P_0>0, \mathbf{t}, \mathbf{p}} \quad & P_0 t_0 && \text{(3.11a)}\\
\text{s.t.} \quad & T - (t_0 + t_1 + t_{21} + t_{22}) \geq 0, && \text{(3.11b)}\\
& E_{\mathrm{s},1}\left(P_0, \mathbf{t}, \mathbf{p}\right) \geq 0, && \text{(3.11c)}\\
& E_{\mathrm{s},2}\left(P_0, \mathbf{t}, \mathbf{p}\right) \geq 0, && \text{(3.11d)}\\
& M_1^+ \leq L_1(\mathbf{t}, \mathbf{p}) \leq I_1, && \text{(3.11e)}\\
& M_2^+ \leq L_2(\mathbf{t}, \mathbf{p}) \leq I_2, && \text{(3.11f)}\\
& t_0 \geq 0, t_1 \geq 0, t_{21} \geq 0, t_{22} \geq 0, && \text{(3.11g)}\\
& p_1 \geq 0, p_{21} \geq 0, p_{22} \geq 0, && \text{(3.11h)}
\end{aligned}$$

where (3.11a) is the objective function for minimizing the AP's transmit energy; (3.11b) is the system latency constraint; (3.11c) and (3.11d) respectively represent the energy harvesting causality constraints for UE$_1$ and UE$_2$; (3.11e) and (3.11f) respectively denote the task allocation constraints for UE$_1$ and UE$_2$; (3.11g) and (3.11h) ensure the non-negativeness of the optimized variables. Note that problem (P3.1) is a non-convex optimization problem in the above form because of the

expressions of $L_1(\mathbf{t}, \mathbf{p})$ and $L_2(\mathbf{t}, \mathbf{p})$, and the product of $P_0 t_0$. Actually, problem (P3.1) can be equivalently transformed into the following min-max problem (P3.2)[3]

$$\begin{aligned} (\mathrm{P3.2}): \min_{P_0>0} \max_{\mathbf{t},\mathbf{p}} \quad & E_{\mathrm{s},1}(\mathbf{t}, \mathbf{p}) + E_{\mathrm{s},2}(\mathbf{t}, \mathbf{p}) \\ \text{s.t.} \quad & (3.11\text{b})\text{–}(3.11\text{h}). \end{aligned} \tag{3.12}$$

However, problem (P3.2) is still non-convex in this form. In order to make this problem solvable and facilitate further analysis, we propose a two-phase method. In the first phase, we solve the inner subproblem with a given P_0 where the sum-energy-saving (SES), i.e., $E_{\mathrm{s},1}(\mathbf{t}, \mathbf{p}) + E_{\mathrm{s},2}(\mathbf{t}, \mathbf{p})$ is maximized under the constraints in (P3.1), referred to as the SES maximization (SESM) problem (P3.3):

$$\begin{aligned} (\mathrm{P3.3}): \max_{\mathbf{t},\mathbf{p}} \quad & E_{\mathrm{s},1}(\mathbf{t}, \mathbf{p}) + E_{\mathrm{s},2}(\mathbf{t}, \mathbf{p}) \\ \text{s.t.} \quad & (3.11\text{b})\text{–}(3.11\text{h}), \end{aligned} \tag{3.13}$$

through which the optimal time and power allocation corresponding to the given P_0 can be obtained. In the second phase, we will find the optimal minimum of P_0 through a bi-section search method. In the following section, we will demonstrate the details of the problem-solving process with the two-phase method.

3.3 Proposed Two-Phase Method

In this section, we focus on designing the two-phase method for the joint power and time allocation of the considered wireless powered cooperation-assisted MEC system. The process of operating the first phase with a given P_0 is presented in Sections 3.3.1 to 3.3.4, where the optimal offloaded data sizes of UEs, the

[3]The proof of verifying the equivalence between problems (P3.1) and (P3.2) will be given in Section 3.3.5 after solving the inner SESM subproblem (P3.3) since the proof needs some results obtained through solving problem (P3.3).

power allocation of UE_1 (in semi-closed form) and UE_2 (in closed form) as well as the optimal time allocation are obtained for each subproblem with a given P_0. Besides, the equivalence between problem (P3.1) and (P3.2) is given in Section 3.3.5. Finally, the second phase is described in Section 3.3.6, where the optimal minimum of $P_0^\star$ is achieved.

3.3.1 Transforming the SESM Problem (P3.3) into Convex

To make the non-convex SESM problem (P3.3) in (3.13) solvable with a given P_0, we first introduce the variables $q_1 = \frac{p_1 t_1}{\nu_1 g_1 P_0}$ and $q_{21} = \frac{p_{21} t_{21}}{\nu_2 g_2 P_0}$. By denoting $\mathbf{q} = [q_1, q_{21}]$, $L_{1,1}(\mathbf{t}, \mathbf{p})$, $L_{1,2}(\mathbf{t}, \mathbf{p})$ and $L_{1,12}(\mathbf{t}, \mathbf{p})$ described in (3.3)–(3.5) can then be re-expressed as functions of $\mathbf{t}$ and $\mathbf{q}$ as

$$L_{1,1}(\mathbf{t}, \mathbf{q}) = t_1 B \log_2 \left(1 + \beta_1 P_0 \frac{q_1}{t_1}\right), \tag{3.14}$$

$$L_{1,2}(\mathbf{t}, \mathbf{q}) = t_{21} B \log_2 \left(1 + \beta_2 P_0 \frac{q_{21}}{t_{21}}\right), \tag{3.15}$$

$$L_{1,12}(\mathbf{t}, \mathbf{q}) = t_1 B \log_2 \left(1 + \beta_{12} P_0 \frac{q_1}{t_1}\right), \tag{3.16}$$

where $\beta_1 = \frac{\nu_1 g_1 h_1}{N_0}$, $\beta_2 = \frac{\nu_2 g_2 h_2}{N_0}$, and $\beta_{12} = \frac{\nu_1 g_1 h_{12}}{N_2}$. Note that the above three functions equal to 0 when $t_1 = 0$, $t_{21} = 0$ and $t_1 = 0$, respectively. Using the property of perspective function [123], it is easily verified that $L_{1,1}(\mathbf{t}, \mathbf{q})$, $L_{1,2}(\mathbf{t}, \mathbf{q})$ and $L_{1,12}(\mathbf{t}, \mathbf{q})$ are all joint concave functions of $\mathbf{t}$ and $\mathbf{q}$. Besides, they are all monotonically increasing functions over each element of (t_1, q_1), (t_{21}, q_{21}) and (t_1, q_1), respectively. Next, we introduce a new variable

$$\overline{L}_1 = \min \left\{ L_{1,1}(\mathbf{t}, \mathbf{q}) + L_{1,2}(\mathbf{t}, \mathbf{q}), L_{1,12}(\mathbf{t}, \mathbf{q}) \right\} \tag{3.17}$$

to replace $L_1(\mathbf{t}, \mathbf{p})$ in problem (P3.3) with two additional convex constraints

$$L_{1,1}(\mathbf{t}, \mathbf{q}) + L_{1,2}(\mathbf{t}, \mathbf{q}) \geq \overline{L}_1, \tag{3.18}$$

and

$$L_{1,12}(\mathbf{t}, \mathbf{q}) \geq \overline{L}_1. \tag{3.19}$$

Thus, the expression of energy saving for UE_1 in the objective function of problem (P3.3) (and its corresponding constraints) has been turned into concave (convex). However, even though we can use a similar variable-changing method to convert $L_2(\mathbf{t}, \mathbf{p})$ into a concave function $L_2(\mathbf{t}, \mathbf{q})$, the corresponding constraint $L_2(\mathbf{t}, \mathbf{q}) \leq I_2$ in (3.11f) is still non-convex. To tackle this issue, we redefine the offloaded data size of UE_2 as an independent variable L_2, and then by defining a convex function

$$g(x) = N_0(2^{\frac{x}{B}} - 1),\ x \geq 0, \tag{3.20}$$

the offloading power p_{22} can be described as a function of L_2 and t_{22} according to (3.6), given by

$$p_{22} = \frac{1}{h_2} g\left(\frac{L_2}{t_{22}}\right). \tag{3.21}$$

Hence, the energy savings for UE_1 and UE_2 with a given P_0 can be rewritten as

$$E_{\text{s},1}\left(\mathbf{t}, \mathbf{q}, \overline{L}_1\right) = \nu_1 g_1 P_0(t_0 - q_1) - (I_1 - \overline{L}_1) C_1 Q_1, \tag{3.22}$$

$$E_{\text{s},2}\left(\mathbf{t}, \mathbf{q}, L_2\right) = \nu_2 g_2 P_0(t_0 - q_{21}) - \frac{t_{22}}{h_2} g\left(\frac{L_2}{t_{22}}\right) - (I_2 - L_2) C_2 Q_2. \tag{3.23}$$

Therefore, the SESM problem (P3.3) can be equivalently reformulated as another SESM problem (P3.4)

$$
\begin{aligned}
(\mathrm{P3.4}): \max_{\mathbf{t},\mathbf{q},\overline{L}_1,L_2} \quad & E_{\mathrm{s},1}\left(\mathbf{t},\mathbf{q},\overline{L}_1\right)+E_{\mathrm{s},2}\left(\mathbf{t},\mathbf{q},L_2\right) && \text{(3.24a)}\\
\text{s.t.} \quad & T-\left(t_0+t_1+t_{21}+t_{22}\right)\geq 0, && \text{(3.24b)}\\
& E_{\mathrm{s},1}\left(\mathbf{t},\mathbf{q},\overline{L}_1\right)\geq 0, && \text{(3.24c)}\\
& E_{\mathrm{s},2}\left(\mathbf{t},\mathbf{q},L_2\right)\geq 0, && \text{(3.24d)}\\
& L_{1,1}(\mathbf{t},\mathbf{q})+L_{1,2}(\mathbf{t},\mathbf{q})\geq \overline{L}_1, && \text{(3.24e)}\\
& L_{1,12}(\mathbf{t},\mathbf{q})\geq \overline{L}_1, && \text{(3.24f)}\\
& M_1^+\leq \overline{L}_1\leq I_1, && \text{(3.24g)}\\
& M_2^+\leq L_2\leq I_2, && \text{(3.24h)}\\
& t_0\geq 0, t_1\geq 0, t_{21}\geq 0, t_{22}\geq 0, && \text{(3.24i)}\\
& q_1\geq 0, q_{21}\geq 0. && \text{(3.24j)}
\end{aligned}
$$

As $g(x)$ is a convex function, its perspective function $t_{22}g(\frac{L_2}{t_{22}})$ is a joint convex function of t_{22} and L_2 considering both the cases of $t_{22}>0$ and $t_{22}=0$ [123]. Therefore, the objective function is concave and all the constraints are convex, constituting a convex optimization problem (P3.4).

3.3.2 Problem-Solving with Lagrange Method

To gain more insights into the solution, we next solve the equivalent SESM problem (P3.4) optimally by leveraging the Lagrange method [123]. The partial Lagrange

function of (P3.4) is defined as

$$\begin{aligned}
&\mathcal{L}(\mathbf{t}, \mathbf{q}, \overline{L}_1, L_2, \eta, \boldsymbol{\lambda}) \\
&\triangleq (1+\lambda_1) E_{\mathrm{s},1}\left(\mathbf{t}, \mathbf{q}, \overline{L}_1\right) + (1+\lambda_2) E_{\mathrm{s},2}\left(\mathbf{t}, \mathbf{q}, L_2\right) \\
&\quad + \eta\left(T - (t_0 + t_1 + t_{21} + t_{22})\right) \\
&\quad + \lambda_3\left(L_{1,1}(\mathbf{t}, \mathbf{q}) + L_{1,2}(\mathbf{t}, \mathbf{q}) - \overline{L}_1\right) \\
&\quad + \lambda_4\left(L_{1,12}(\mathbf{t}, \mathbf{q}) - \overline{L}_1\right),
\end{aligned} \tag{3.25}$$

where $\eta \geq 0$ and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_4] \succeq \mathbf{0}$ ($\succeq$ denotes the componentwise inequality) consist of the Lagrange multipliers associated with the constraints (3.24b) and (3.24c)-(3.24f) in problem (P3.4), respectively. In order to facilitate the analysis in the sequel, we define another two functions

$$f(x) = \ln(1+x) + \frac{1}{1+x} - 1,\ x \geq 0, \tag{3.26}$$

$$h(x) = g(x) - xg'(x),\ x \geq 0, \tag{3.27}$$

where $g'(x)$ denotes the first-order derivative of $g(x)$, and thus the following two lemmas are established.

Lemma 3.1. *$f(x)$ is a monotonic increasing function of $x \geq 0$ with $f(0) = 0$. Given $C > 0$, there exists a unique positive solution for equation $f(x) = C$, given by*

$$x^* = -\left(1 + \frac{1}{W_0(-e^{(-(C+1))})}\right), \tag{3.28}$$

where $W_0(z)$, defined as the solution for equation $W_0(z)e^{W_0(z)} = z$ [124], is the principal branch of the lambert W function, and e is the base of the natural

logarithm.

Proof. It is easy to verify that $f(x)$ is a monotonic increasing function for $x \geq 0$ with $f(0) = 0$ by simply deriving its first-order derivative. Hence, the equation $f(x) = C$ with $C > 0$ has a unique positive solution. Through derivation, $f(x) = C$ can be equivalently expressed as

$$-\frac{1}{1+x}e^{-\frac{1}{1+x}} = -e^{-(C+1)} \in (-e^{-1}, 0). \tag{3.29}$$

By using the definition and property of Lambert function [124], we obtain the solution $x^* = -\left(1 + \frac{1}{W_0(-e^{(-(C+1))})}\right) > 0$, where $W_0(-e^{(-(C+1))}) \in (-1, 0)$. □

Lemma 3.2. *$h(x)$ is a monotonic decreasing function of $x \geq 0$ with $h(0) = 0$. Given $G < 0$, there exists a unique positive solution for equation $h(x) = G$, given by*

$$x^* = \frac{B}{\ln 2}\left[W_0\left(\frac{G/N_0 + 1}{-e}\right) + 1\right]. \tag{3.30}$$

Proof. Similar to **Lemma** 3.1, by deriving the first-order derivative of $h(x)$, we can verify that $h(x)$ is a monotonic decreasing function of $x \geq 0$ with $h(0) = 0$. Hence, the equation $h(x) = G$ with $G < 0$ has a unique positive solution. Through derivation, $h(x) = G$ can be equivalently expressed as

$$\left(\frac{\ln 2}{B}x - 1\right)e^{(\frac{\ln 2}{B}x - 1)} = \frac{G/N_0 + 1}{-e}. \tag{3.31}$$

Therefore, we obtain $x^* = \frac{B}{\ln 2}\left[W_0\left(\frac{G/N_0+1}{-e}\right) + 1\right] > 0$ by using the definition and property of Lambert function [124], where $W_0\left(\frac{G/N_0+1}{-e}\right) > W_0(-e^{-1}) = -1$. □

We first assume that problem (P3.4) is feasible with the given AP's energy

transmit power P_0, and let $(\mathbf{t}^*, \mathbf{q}^*, \overline{L}_1^*, L_2^*)$ denote the optimal solution of problem (P3.4) and η^*, $\boldsymbol{\lambda}^*$ denote the optimal Lagrange multipliers. Then applying the Karush-Kuhn-Tucker (KKT) conditions [123], the following necessary and sufficient conditions can be obtained:

$$\frac{\partial \mathcal{L}}{\partial t_0^*} = (1+\lambda_1^*)\nu_1 g_1 P_0 + (1+\lambda_2^*)\nu_2 g_2 P_0 - \eta^* = 0, \tag{3.32}$$

$$\frac{\partial \mathcal{L}}{\partial t_1^*} = \frac{B\lambda_3^*}{\ln 2} f\left(\beta_1 P_0 \frac{q_1^*}{t_1^*}\right) + \frac{B\lambda_4^*}{\ln 2} f\left(\beta_{12} P_0 \frac{q_1^*}{t_1^*}\right) - \eta^* \begin{cases} < 0, t_1^* = 0, \\ = 0, t_1^* > 0, \end{cases} \tag{3.33}$$

$$\frac{\partial \mathcal{L}}{\partial t_{21}^*} = \frac{B\lambda_3^*}{\ln 2} f\left(\beta_2 P_0 \frac{q_{21}^*}{t_{21}^*}\right) - \eta^* \begin{cases} < 0, t_{21}^* = 0, \\ = 0, t_{21}^* > 0, \end{cases} \tag{3.34}$$

$$\frac{\partial \mathcal{L}}{\partial t_{22}^*} = -(1+\lambda_2^*)\frac{1}{h_2} h\left(\frac{L_2^*}{t_{22}^*}\right) - \eta^* \begin{cases} < 0, t_{22}^* = 0, \\ = 0, t_{22}^* > 0, \end{cases} \tag{3.35}$$

$$\frac{\partial \mathcal{L}}{\partial q_1^*} = -(1+\lambda_1^*)\nu_1 g_1 P_0 + \frac{B}{\ln 2} \times \left(\frac{\lambda_3^* \beta_1 P_0}{1+\beta_1 P_0 \frac{q_1^*}{t_1^*}} + \frac{\lambda_4^* \beta_{12} P_0}{1+\beta_{12} P_0 \frac{q_1^*}{t_1^*}}\right) \begin{cases} < 0, q_1^* = 0, \\ = 0, q_1^* > 0, \end{cases} \tag{3.36}$$

$$\frac{\partial \mathcal{L}}{\partial q_{21}^*} = -(1+\lambda_2^*)\nu_2 g_2 P_0 + \frac{B}{\ln 2}\left(\frac{\lambda_3^* \beta_2 P_0}{1+\beta_2 P_0 \frac{q_{21}^*}{t_{21}^*}}\right) \begin{cases} < 0, q_{21}^* = 0, \\ = 0, q_{21}^* > 0, \end{cases} \tag{3.37}$$

$$\frac{\partial \mathcal{L}}{\partial \overline{L}_1^*} = (1+\lambda_1^*)C_1 Q_1 - \lambda_3^* - \lambda_4^* \begin{cases} < 0, \overline{L}_1^* = M_1^+, \\ = 0, \overline{L}_1^* \in (M_1^+, I_1), \\ > 0, \overline{L}_1^* = I_1, \end{cases} \tag{3.38}$$

$$\frac{\partial \mathcal{L}}{\partial L_2^*} = (1+\lambda_2^*)\left[C_2 Q_2 - \frac{1}{h_2} g'\left(\frac{L_2^*}{t_{22}^*}\right)\right] \begin{cases} < 0, L_2^* = M_2^+, \\ = 0, L_2^* \in (M_2^+, I_2), \\ > 0, L_2^* = I_2, \end{cases} \tag{3.39}$$

$$\eta^* \left(T - (t_0^* + t_1^* + t_{21}^* + t_{22}^*)\right) = 0, \tag{3.40}$$

$$\lambda_1^* E_{\mathrm{s},1} \left(\mathbf{t}^*, \mathbf{q}^*, \overline{L}_1^*\right) = 0, \tag{3.41}$$

$$\lambda_2^* E_{\mathrm{s},2} \left(\mathbf{t}^*, \mathbf{q}^*, L_2^*\right) = 0, \tag{3.42}$$

$$\lambda_3^* \left(L_{1,1}(\mathbf{t}^*, \mathbf{q}^*) + L_{1,2}(\mathbf{t}^*, \mathbf{q}^*) - \overline{L}_1^*\right) = 0, \tag{3.43}$$

$$\lambda_4^* \left(L_{1,12}(\mathbf{t}^*, \mathbf{q}^*) - \overline{L}_1^*\right) = 0. \tag{3.44}$$

Note that $t_0^* + t_1^* + t_{21}^* + t_{22}^* = T$ must hold; otherwise, we can always allocate the remaining time to t_0^* to further increase the energy saving of the two users, and thus $\eta^* > 0$ holds for sure. Furthermore, the following lemma describes an important result concerning $\mathbf{t}^*$, $\mathbf{q}^*$ and $\overline{L}_1^*$:

Lemma 3.3. *The optimal time and power allocation* $(\mathbf{t}^*, \mathbf{q}^*)$ *ensures the following property of UE*$_1$*'s offloaded data size,* $\overline{L}_1^*$.

$$\overline{L}_1^* = L_{1,1}(\mathbf{t}^*, \mathbf{q}^*) + L_{1,2}(\mathbf{t}^*, \mathbf{q}^*) \leq L_{1,12}(\mathbf{t}^*, \mathbf{q}^*). \tag{3.45}$$

Proof. According to the definition of $g(x)$, $h(x)$, and condition (3.35), we know that

$$\frac{\partial(t_{22} g(\frac{L_2}{t_{22}}))}{\partial t_{22}} = h\left(\frac{L_2}{t_{22}}\right) < 0 \text{ for } t_{22} > 0, \tag{3.46}$$

which indicates that $t_{22} g(\frac{L_2}{t_{22}})$ is a monotonically decreasing function of t_{22}. It is easy to prove that the inequality $L_{1,1}(\mathbf{t}^*, \mathbf{q}^*) < L_{1,12}(\mathbf{t}^*, \mathbf{q}^*)$ always holds for the considered case of $h_1 < h_{12}$, as indicated in footnote 1 of this chapter. If $L_{1,1}(\mathbf{t}^*, \mathbf{q}^*) + L_{1,2}(\mathbf{t}^*, \mathbf{q}^*) > L_{1,12}(\mathbf{t}^*, \mathbf{q}^*) \geqslant \overline{L}_1^*$ holds, we can always allocate part of t_{21}^* to t_{22}^* while maintaining the same $\overline{L}_1^*$, L_2^*, $\mathbf{q}^*$, t_0^*, t_1^* and the sum of t_{21}^*, t_{22}^*,

which will decrease $L_{1,2}(\mathbf{t}^*, \mathbf{q}^*)$ until the equality holds. This operation will result in an increased $E_{\mathrm{s},2}\left(\mathbf{t}^*, \mathbf{q}^*, L_2^*\right)$ (expression (3.23)) by decreasing $t_{22}^* g(\frac{L_2^*}{t_{22}^*})$ without reducing $E_{\mathrm{s},1}(\mathbf{t}^*, \mathbf{q}^*, \overline{L}_1^*)$ (expression (3.22)), and thus will increase the objective function of problem (P3.4) while satisfy all the constraints. Hence, expression (3.45) always holds with the optimal solution of problem (P3.4). □

Remark 3.1. *(Intuitive Explanation).* **Lemma** 3.3 sheds light on the fact that the optimal offloaded data size of UE_1, i.e., $\overline{L}_1^*$ should be the sum of the decoded data sizes at the AP, i.e., $(L_{1,1}(\mathbf{t}^*, \mathbf{q}^*) + L_{1,2}(\mathbf{t}^*, \mathbf{q}^*))$ rather than $L_{1,12}(\mathbf{t}^*, \mathbf{q}^*)$, which simplifies the expression of $\overline{L}_1$ compared with that in expression (3.17).

Based on the result of **Lemma** 3.3, we can derive that $\lambda_3^* > 0$ and $\lambda_4^* = 0$. Furthermore, for $\mathbf{t}^* \succ 0$ and $\mathbf{q}^* \succ 0$, it can be derived from the KKT conditions (3.33), (3.34) and the result of **Lemma** 3.1 that

$$\beta_1 \frac{q_1^*}{t_1^*} = \beta_2 \frac{q_{21}^*}{t_{21}^*} = -\frac{1}{P_0}\left(1 + \left(W_0\left(-e^{-\left(\frac{\eta^* \ln 2}{\lambda_3^* B}+1\right)}\right)\right)^{-1}\right). \tag{3.47}$$

Moreover, through the KKT conditions (3.36) and (3.37), we can respectively derive that

$$\beta_1 \frac{q_1^*}{t_1^*} = \frac{\lambda_3^* B \beta_1}{(1+\lambda_1^*)\nu_1 g_1 P_0 \ln 2} - \frac{1}{P_0}, \tag{3.48}$$

$$\beta_2 \frac{q_{21}^*}{t_{21}^*} = \frac{\lambda_3^* B \beta_2}{(1+\lambda_2^*)\nu_2 g_2 P_0 \ln 2} - \frac{1}{P_0}. \tag{3.49}$$

Based on (3.47)-(3.49), we obtain that $(1+\lambda_1^*)\nu_1 g_1 P_0 = \frac{\beta_1}{\beta_2}(1+\lambda_2^*)\nu_2 g_2 P_0$. Combining the condition (3.32), the optimal Lagrange multipliers have the following property:

$$(1+\lambda_k^*)\nu_k g_k P_0 = \frac{\beta_k \eta^*}{\beta_1 + \beta_2},\ k \in \{1,2\}. \tag{3.50}$$

Hence, by substituting (3.50) into (3.48) and (3.49), we obtain

$$\beta_1 \frac{q_1^*}{t_1^*} = \beta_2 \frac{q_{21}^*}{t_{21}^*} = \frac{B\lambda_3^*(\beta_1+\beta_2)}{\eta^* \ln 2} - \frac{1}{P_0}. \tag{3.51}$$

Based on these results, the optimal resource allocation of problem (P3.4) for a given feasible P_0 is characterized in the following sections.

3.3.3 Optimal Offloading Decisions with Power Allocation

First, we define an offloading priority indicator for UE_k as

$$\mu_k \triangleq \frac{Bh_kC_kQ_k}{N_0 \ln 2},\ k \in \{1,2\}. \tag{3.52}$$

Note that μ_k depends on the corresponding variables quantifying uplink offloading channel (h_k), local computing overhead (C_kQ_k), and it is a monotonically increasing function of h_k, C_k and Q_k. The relationship between the optimal offloaded data size and power allocation for each user with the corresponding offloading priority indicator is shown in the following theorem.

Theorem 3.1. *(Optimal Cooperative Computation Offloading Decisions with Power Allocation).*

1) If $M_1^+ > 0$ or $\mu_1 \geq (\beta_1+\beta_2)P_0/z^*$, the optimal $\overline{L}_1^*$, p_1^* and p_{21}^* (all in semi-closed form) can be expressed as

$$\overline{L}_1^* \begin{cases} = M_1^+, & \mu_1 < \dfrac{(\beta_1+\beta_2)P_0}{z^*}, \\ \in (M_1^+, I_1), & \mu_1 = \dfrac{(\beta_1+\beta_2)P_0}{z^*}, \\ = I_1, & \mu_1 > \dfrac{(\beta_1+\beta_2)P_0}{z^*}, \end{cases} \tag{3.53}$$

$$p_1^* = \frac{N_0}{h_1}\left(\frac{(\beta_1+\beta_2)P_0}{z^*} - 1\right) > 0, \tag{3.54}$$

$$p_{21}^* = \frac{N_0}{h_2}\left(\frac{(\beta_1+\beta_2)P_0}{z^*} - 1\right) > 0, \tag{3.55}$$

in which z^* *is the unique solution of the equation given by* $e^{\left(\frac{1}{(\beta_1+\beta_2)P_0}-1\right)z} - \frac{e}{(\beta_1+\beta_2)P_0}z = 0$ *on the specific range of* $z \in (0, (\beta_1+\beta_2)P_0)$.

If $M_1^+ = 0$ *and* $\mu_1 < (\beta_1+\beta_2)P_0/z^*$, *it is optimal to set* $\overline{L}_1^* = 0$, $p_1^* = 0$, *and* $p_{21}^* = 0$.

2) If $M_2^+ > 0$ *or* $\rho(\mu_2) \geq (\beta_1+\beta_2)P_0$, *the optimal* L_2^* *and* p_{22}^* *(all in closed form) are given by*

$$L_2^* \begin{cases} = M_2^+, & \rho(\mu_2) < (\beta_1+\beta_2)P_0, \\ \in (M_2^+, I_2), & \rho(\mu_2) = (\beta_1+\beta_2)P_0, \\ = I_2, & \rho(\mu_2) > (\beta_1+\beta_2)P_0, \end{cases} \tag{3.56}$$

$$p_{22}^* = \frac{1}{h_2} g\left(\frac{B}{\ln 2}\left[W_0\left(\frac{(\beta_1+\beta_2)P_0 - 1}{e}\right) + 1\right]\right) > 0, \tag{3.57}$$

where $\rho(\mu_2) \triangleq \mu_2 \ln \mu_2 - \mu_2 + 1$.

If $M_2^+ = 0$ *and* $\rho(\mu_2) < (\beta_1+\beta_2)P_0$, *it is optimal to set* $L_2^* = 0$ *and* $p_{22}^* = 0$.

Proof. See Appendix A.1. □

Lemma 3.4. *(Quick Offloading Decisions for the Minimum Offloaded Data Size of UE*$_1$ *and UE*$_2$*).* When $\mu_1 \leq 1$ (or $\mu_2 \leq 1$), the optimal offloaded data size for UE$_1$ (or UE$_2$) is the minimum, i.e., $\overline{L}_1^* = M_1^+$ (or $L_2^* = M_2^+$). In these two cases, we can get the optimal $\overline{L}_1^*$ (or L_2^*) just according to the value of μ_1 (or μ_2) without making comparisons as in (3.53) (or (3.56)).

Proof. Based on the expression of $\frac{\partial \mathcal{L}}{\partial \overline{L}_1^*}$ in (A.1.3) of Appendix A.1 and the range

of $z^* \in (0, (\beta_1 + \beta_2)P_0)$, we can verify that $\frac{\partial \mathcal{L}}{\partial \overline{L}_1^*} < 0$ when $\mu_1 \leq 1$, and thus $\overline{L}_1^* = M_1^+$. As for UE$_2$, $\frac{L_2^*}{t_{22}^*} > \frac{B}{\ln 2} \ln \mu_2$ always holds when $\mu_2 \leq 1$ and $L_2^* > 0$, which is equivalent to $\frac{\partial \mathcal{L}}{\partial L_2^*} < 0$ according to the proof of **Theorem** 3.1, and thus $L_2^* = M_2^+$, which completes the proof. □

Remark 3.2. *(Whether Computation Offloading is Necessary?).* According to **Theorem** 3.1, it is easy to note that the offloading decision and power allocation of each user depend on their corresponding offloading priority indicator μ_k as well as the minimum required offloaded data size M_k^+, $k \in \{1, 2\}$. If $M_1^+ = 0$ and $\mu_1 < (\beta_1 + \beta_2)P_0/z^*$, then operating the whole computation task locally is optimal for UE$_1$; otherwise computation offloading is required. Similarly, if $M_2^+ = 0$ and $\rho(\mu_2) < (\beta_1 + \beta_2)P_0$, then fulfilling the whole computation task locally is optimal for UE$_2$; otherwise computation offloading is necessary.

Remark 3.3. *(Effects of Parameters on the Offloading Priority).* It is easy to note that $\rho(\mu_2)$ is a monotonic increasing function of μ_2 for $\mu_2 > 1$ (as for $\mu_2 \leq 1$, $L_2^* = M_2^+$ according to **Lemma** 3.4), and thus it also monotonically increases with parameters C_2, Q_2, and h_2 in this case, according to the monotonicity rule of compound function. The results in **Theorem** 3.1 show that the optimal offloaded data sizes for the two cooperative users UE$_k$, $k \in \{1, 2\}$ grow with increasing μ_k, which is consistent with the intuition that more resources should be scheduled to computation offloading when users have good channels (i.e., large h_k) or endure high local computing energy consumption (i.e., large C_k and Q_k), so as to save energy.

Remark 3.4. *(Binary Structure of the Offloading Decisions for Two Cooperative Users).* **Theorem** 3.1 reveals that the optimal offloading decisions for both UE$_1$ and UE$_2$ have a similar threshold-based structure when computation offloading

saves energy. Moreover, since the exact cases of $\mu_1 = (\beta_1 + \beta_2)P_0/z^*$ in (3.53) and $\rho(\mu_2) = (\beta_1 + \beta_2)P_0$ in (3.56) rarely occur in practice, the optimal offloading decisions have a binary structure for both cooperative users.

Remark 3.5. *(Effects of Parameters on the Thresholds of the Offloading Decisions).* The same item in the thresholds of the offloading decisions for the two users in **Theorem** 3.1, i.e., $(\beta_1 + \beta_2) = (\nu_1 g_1 h_1 + \nu_2 g_2 h_2)/N_0$, reflects the energy harvesting potentials of the two users (i.e., $\nu_1 g_1$ and $\nu_2 g_2$) and the quality of uplink offloading channels for the users (i.e., h_1 and h_2), which demonstrates the effect of user cooperation that either user's offloading decision is affected by the other user's energy-harvesting ability and offloading-channel quality.

Lemma 3.5. *For the case of $\overline{L}_1^* > 0$, the optimal transmit rates of UE$_1$ and UE$_2$ for offloading UE$_1$'s input data are same, which is expressed as*

$$r_{1,1}(\mathbf{p}^*) = r_{1,2}(\mathbf{p}^*) = B \log_2 \frac{(\beta_1 + \beta_2)P_0}{z^*}. \tag{3.58}$$

Proof. It is easy to verify the result in **Lemma** 3.5 by substituting the optimal transmit power in (3.54) and (3.55) into the expressions of $r_{1,1}(\mathbf{p})$ and $r_{1,2}(\mathbf{p})$ in (3.3) and (3.4), respectively. □

3.3.4 Optimal Energy-Efficient Time Allocation

Using **Theorem** 3.1, we have obtained the optimal offloaded data size, i.e., $(\overline{L}_1^*, L_2^*)$ and the optimal power allocation, i.e., $\mathbf{p}^* = (p_1^*, p_{21}^*, p_{22}^*)$, for the SESM problem (P3.3) under a given feasible P_0. In this subsection, we focus on obtaining the corresponding optimal time allocation, i.e., $\mathbf{t}^* = (t_0^*, t_1^*, t_{21}^*, t_{22}^*)$, which is summarised in **Theorem** 3.2.

Theorem 3.2. *(Optimal Time Allocation for WPT and Cooperative Computation Offloading).*

1) The optimal time allocation for offloading UE$_2$'s input data is given by

$$t_{22}^* = \frac{\ln 2 \times L_2^*}{B\left[W_0\left(\frac{(\beta_1+\beta_2)P_0-1}{e}\right)+1\right]}. \tag{3.59}$$

2) The optimal WPT duration time can be derived as

$$t_0^* = \begin{cases} T - t_{22}^* - \overline{L}_1^*/r_{1,1}(\mathbf{p}^*), & \overline{L}_1^* > 0, \\ T - t_{22}^*, & \overline{L}_1^* = 0. \end{cases} \tag{3.60}$$

3) The optimal time allocation for offloading UE$_1$'s input data, i.e., (t_1^*, t_{21}^*) can be expressed as

$$\begin{cases} t_1^* = \dfrac{\overline{L}_1^*}{r_{1,12}(\mathbf{p}^*)}, \\ t_{21}^* = \dfrac{\overline{L}_1^*}{r_{1,1}(\mathbf{p}^*)} - t_1^*, \end{cases} \tag{3.61}$$

where $(t_1^*, t_{21}^*) = (0,0)$ when $\overline{L}_1^* = 0$.

Proof. See Appendix A.2. □

Remark 3.6. *(Time Allocation versus UE$_1$'s Offloaded Data).* From 3) of **Theorem** 3.2, we can easily see that if local computing is preferred to complete the whole computation task of UE$_1$, i.e., $\overline{L}_1^* = 0$, no time will be allocated to UE$_1$ for offloading as well as UE$_2$ for cooperatively offloading UE$_1$'s task bits, and thus $(t_1^*, t_{21}^*) = (0,0)$. All the remaining time of the slot except the time used for offloading UE$_2$'s task-input data, i.e., $t_0^* = T - t_{22}^*$, will be utilized for WPT so as to maximize the saved energy at the UEs as shown in 2) of **Theorem** 3.2. For the case with $\overline{L}_1^* > 0$, positive time allocation for (t_1^*, t_{21}^*) is necessary, and the equality

of $t_1^* + t_{21}^* = \frac{\overline{L}_1^*}{r_{1,1}(\mathbf{p}^*)} = \frac{\overline{L}_1^*}{r_{1,2}(\mathbf{p}^*)}$ is satisfied, due the facts obtained from Lemma 3.3 and Lemma 3.5. Hence, the remaining time of the slot, i.e., $t_0^* = T - t_{22}^* - t_1^* - t_{21}^*$ will be used for energy harvesting.

Remark 3.7. *(Time Allocation versus UE$_2$'s Offloaded Data).* Similarly, if local computing is preferred to complete the whole computation task of UE$_2$, i.e., $L_2^* = 0$, no time will be allocated for offloading UE$_2$'s task-input data, i.e., $t_{22}^* = 0$ as seen from 1) of **Theorem** 3.2. In contrast, $t_{22}^* > 0$ if $L_2^* > 0$, which also depends on the offloading power of UE$_2$, i.e., p_{22}^* given in (3.57).

Remark 3.8. *(Relationship between Time Allocation and User Cooperation).* The **Theorem** 3.2 in combination with the power allocation in **Theorem** 3.1 further show that the optimal time allocation for UEs' computation offloading and WPT are highly related to the strategy of user cooperation. Based on the power allocation in **Theorem** 3.1, we know that $r_{1,1}(\mathbf{p}^*) = r_{1,2}(\mathbf{p}^*) = B\log_2\frac{(\beta_1+\beta_2)P_0}{z^*}$ and $r_{1,12}(\mathbf{p}^*) = B\log_2(1+\frac{N_0h_{12}}{h_2}\left(1+\frac{(\beta_1+\beta_2)P_0}{z^*}-1\right)$. We can see that the value of $(\beta_1+\beta_2) = (\nu_1 g_1 h_1 + \nu_2 g_2 h_2)/N_0$ plays an important role in determining the values of t_1^*, t_{21}^* and/or t_{22}^* for computation offloading when $\overline{L}_1^* > 0$ and/or $L_2^* > 0$, and further affect the value of t_0^* for WPT. As discussed in Remark 3.5, the value of $(\beta_1 + \beta_2)$ reflects the energy harvesting potentials of the two users (i.e., $\nu_1 g_1$ and $\nu_2 g_2$) and the quality of uplink offloading channels for the users (i.e., h_1 and h_2), which further indicates the effect of cooperation among near-far users that either user's allocated time for computation offloading is affected by the other user's energy-harvesting ability and offloading-channel quality, and then have an influence on the time utilized for energy harvesting.

3.3.5 The Equivalence Between Problem (P3.1) and Problem (P3.2)

In this part, we proceed to show the equivalence between the original APTEM problem (P3.1) and the min-max problem (P3.2). First, an important property of the optimal WPT duration time t_0^* is given in the following **Lemma** 3.6.

Lemma 3.6. *The optimal WPT duration time t_0^* is a monotonic non-decreasing function of P_0.*

Proof. See Appendix A.3. □

Remark 3.9. *(The Effect of P_0 and t_0 on Maximizing SES).* The result of **Lemma** 3.6 shows that t_0^* is small when P_0 is relatively small, since in this case the extra energy harvested by increasing t_0 cannot compensate the extra energy consumed by reducing the time for computation offloading (i.e., $T - t_0$), leading to a smaller energy saving of both users. On the contrary, when P_0 becomes large, t_0^* increases accordingly to obtain more sum-energy saving.

Theorem 3.3. *The APTEM problem (P3.1) is equivalent to the min-max problem (P3.2).*

Proof. We first introduce a transitional problem (P3.5), denoted as the AP's transmit power minimization (APTPM) problem

$$\begin{aligned} (\mathrm{P3.5}): \min_{P_0>0,\mathbf{t},\mathbf{p}} \quad & P_0, \\ \text{s.t.} \quad & \text{(3.11b)–(3.11h)}. \end{aligned} \tag{3.62}$$

In the sequel, we first try to prove the equivalence between problem (P3.2) and problem (P3.5), and then show the equivalence of problem (P3.5) and problem

(P3.1) to finally verify the theorem.

Problem (P3.5) is a general problem for minimizing the WPT transmit power P_0 by jointly optimizing P_0, $\mathbf{t}$ and $\mathbf{p}$, while problem (P3.2) gives a specific method for obtaining the minimum P_0. Problem (P3.2) is solved by a two-phase method where the minimum $P_0^\star$ can be obtained through a one-dimensional (bi-section) search by solving problem (P3.3) (or P3.4) with each given P_0. It is easy to understand that if we assume the given P_0 is the minimum $P_0^\star$, then the optimal $\mathbf{t}^\star$ and $\mathbf{p}^\star$ of problem (P3.5) can be obtained by solving the SESM problem (P3.3) with the given $P_0^\star$. If we find the minimum given $P_0^\star$ that maximizes the sum-energy-saving with all the constraints being satisfied through a bi-section search, then the obtained $(P_0^\star, \mathbf{t}^\star, \mathbf{p}^\star)$, i.e., the optimal solution of problem (P3.2), is actually the joint-optimal solution of problem (P3.5). Hence, we can say that problem (P3.2) and problem (P3.5) are equivalent for obtaining the joint-optimal solution $(P_0^\star, \mathbf{t}^\star, \mathbf{p}^\star)$.

According to the result of **Lemma** 3.6, the optimal WPT duration time of the SESM problem (P3.3), i.e., t_0^*, is a monotonic non-decreasing function of P_0, which indicates that $P_0 t_0^*(P_0)$ is a monotonic increasing function of P_0. Hence, we can conclude that the minimum P_0 of the APTPM problem (P3.5) is same as the optimal P_0 for minimizing $P_0 t_0$ in the original APTEM problem (P3.1), which means that problem (P3.1) and problem (P3.5) are equivalent, finally proving the equivalence between problem (P3.1) and problem (P3.2). This indicates that when the minimum feasible $P_0^\star$ is used in problem (P3.3) (or P3.4), the obtained maximum sum-energy saving reaches its minimum with respect to P_0. □

3.3.6 Optimal Resource Allocation for Obtaining AP's Minimum Energy Transmit Power

In this section, we will discuss the second phase of solving problem (P3.2) to obtain the AP's minimum energy transmit power $P_0^\star$. It is easy to note that with a larger feasible P_0, as extra $\triangle P_0 > 0$ is available, the feasible region of problem (P3.3) (or P3.4) will be larger as well, and thus more extra energy, at least $v_1 g_1 \triangle P_0 t_0 + v_2 g_2 \triangle P_0 t_0$ will be saved, which means that the maximum SES obtained by problem (P3.3) (or P3.4) is a monotonic increasing function of P_0 as long as problem (P3.3) (or P3.4) is feasible. Hence, the minimum $P_0^\star$ of the original APTEM problem (P3.1) can be obtained through a bi-section search of P_0.

As a matter of fact, the optimal time allocation parameters should satisfy the latency constraint (3.7). Note that t_{22}^* monotonically decreases with P_0 (as verified in the Appendix A.3: proof of **Lemma** 3.6), and thus a lower bound of P_0, denoted as P_0^L, can be obtained by solving the equation $t_{22}^*(P_0) = T$ with $L_2^* = M_2$. Based on this P_0^L, we can further obtain a proper upper bound of P_0, denoted as P_0^U, which should make problem (P3.4) feasible and lead to positive energy savings for both of the users. The optimal $P_0^\star$ must be in the range of (P_0^L, P_0^U), and the following lemma shows a property of $P_0^\star$ which gives a stopping criterion of the bi-section search.

Lemma 3.7. *When the minimum feasible $P_0^\star$ is used in problem (P3.3) (or P3.4), at least one of the two users should use up all its harvested energy, i.e., $E_{\mathrm{s},1}^*(P_0^\star) = 0$ or $E_{\mathrm{s},2}^*(P_0^\star) = 0$.*

Proof. The above lemma can be proved by the method of contradiction. If both

$E^*_{\mathrm{s},1}(P_0^\star) > 0$ and $E^*_{\mathrm{s},2}(P_0^\star) > 0$ hold, then at least

$$\triangle P_0 = \min\left\{\frac{E^*_{\mathrm{s},1}(P_0^\star)}{\nu_1 g_1 t_0^*}, \frac{E^*_{\mathrm{s},2}(P_0^\star)}{\nu_2 g_2 t_0^*}\right\} > 0 \tag{3.63}$$

can be reduced to minimize P_0, which will make $E^*_{\mathrm{s},1}(P_0^\star - \triangle P_0) = 0$ or $E^*_{\mathrm{s},2}(P_0^\star - \triangle P_0) = 0$. □

3.3.7 Algorithm Summary

The whole process of solving the original APTEM problem (P3.1) is summarized in **Algorithm** 3.1, where the final optimal $P_0^\star$ and the corresponding offloaded data size $(\overline{L}_1^\star, L_2^\star)$, and power-time allocation $(\mathbf{p}^\star, \mathbf{t}^\star)$ can all be obtained.

Remark 3.10. *(Low-complexity Algorithm).* Through implementing **Algorithm** 3.1, the optimal solutions of the original APTEM problem (P3.1) can be obtained with closed or semi-closed form by substituting the optimal $P_0^\star$ into **Theorem** 3.1 and **Theorem** 3.2. At most two tiers of one-dimensional (bi-section) search are needed to execute **Algorithm** 3.1. The inner tier one is for obtaining z^* in **Theorem** 1-1) and the outer tier one is for acquiring the optimal $P_0^\star$ following the step 2-step 15. Therefore, the complexity of **Algorithm** 3.1 is at most with the order of $\mathcal{O}(1)\ln(1/\sigma)\ln(1/\delta)$, where $\sigma, \delta > 0$ denote the computational accuracies of the two tiers of one-dimensional search. Compared with the traditional block-coordinate descending algorithm where iterative optimization should be operated, the proposed **Algorithm** 3.1 is of much lower complexity.

Algorithm 3.1 Joint Power and Time Allocation Algorithm for Solving the APTEM Problem (P3.1)

1: **Input**: ν_k, g_k, h_k, f_k, (I_k, C_k, T), $Q_k = \kappa_k f_k^2$, $M_k = I_k - f_k T/C_k$, $k \in \{1, 2\}$, and $\omega > 1$, $\delta > 0$, B, T, N_0, h_{12};

2: Initialize $P_0^U = P_0^L$, $\theta = 0$, where $\theta \in \{0, 1\}$ is an indicator for the feasibility of problem (P3.4);

3: **while** $\theta = 0$ **do**

4: Set $P_0^U = \omega P_0^U$, and then obtain the corresponding $(\overline{L}_1^*, L_2^*, \mathbf{p}^*)$ and $\mathbf{t}^*$ according to **Theorem** 3.1 and **Theorem** 3.2;

5: Calculating $E_{\mathrm{s},1}^*(P_0^U)$ and $E_{\mathrm{s},2}^*(P_0^U)$ according to (3.10);

6: **if** $t_0^*(P_0^U) > 0$, $E_{\mathrm{s},1}^*(P_0^U) > 0$, and $E_{\mathrm{s},2}^*(P_0^U) > 0$, **then** $\theta = 1$;

7: **else** $\theta = 0$;

8: **end if**

9: **end while**

10: **while** $P_0^U - P_0^L > \delta$ **do**

11: Set $P_0 = (P_0^L + P_0^U)/2$, and then obtain the corresponding $(\overline{L}_1^*, L_2^*, \mathbf{p}^*)$ and $\mathbf{t}^*$ according to **Theorem** 3.1 and **Theorem** 3.2;

12: Calculating $E_{\mathrm{s},1}^*(P_0)$ and $E_{\mathrm{s},2}^*(P_0)$ according to (3.10);

13: **if** $t_0^*(P_0) > 0$, $E_{\mathrm{s},1}^*(P_0) > 0$, and $E_{\mathrm{s},2}^*(P_0) > 0$, **then** $P_0^U = P_0$;

14: **elseif** $t_0^*(P_0) \leq 0$ or $E_{\mathrm{s},1}^*(P_0) < 0$ or $E_{\mathrm{s},2}^*(P_0) < 0$, **then** $P_0^L = P_0$;

15: **else** break;

16: **end if**

17: **end while**

18: **Output**: $P_0^\star = P_0$, and the corresponding $\overline{L}_1^\star$, $L_2^\star$, $\mathbf{t}^\star = (t_0^\star, t_1^\star, t_{21}^\star, t_{22}^\star)$, $\mathbf{p}^\star = (p_1^\star, p_{21}^\star, p_{22}^\star)$.

3.4 Numerical Results

In this section, the performance of the proposed wireless powered computation offloading scheme with user cooperation by jointly optimizing AP's energy transmit power, UEs' offloading power, and time allocation, is investigated by computer simulations. We will refer to our proposed scheme as "UC-JOPT" in the figures for comparison. Also, we include the results of the following two baselines:

1. A simplified wireless powered computation offloading scheme with user cooperation where UE_1 and UE_2 use equal transmit time to offload UE_1's input data ("UC-ET" for short). In this scheme, $\mathbf{p}$, $\overline{L}_1$, L_2, t_{22} and t_1 are assigned according to the optimal solutions obtained from **Theorem** 3.1 and **Theorem** 3.2. As for t_{21}, UE_2 chooses to use the same time duration as t_1 to relay UE_1's input-data information, i.e., $t_{21} = t_1$, and thus $t_0 = T - t_1 - t_{21} - t_{22}$, which is suboptimal when compared with the optimal resource allocation obtained in the proposed UC-JOPT scheme.

2. Wireless powered computation offloading scheme with inactive user cooperation by letting $t_{21} = 0$ and $t_1 = \overline{L}_1^*/r_{1,1}(\mathbf{p}^*)$ ("IUC" for short).

The simulation settings are set as follows unless specified otherwise. The bandwidth and the time block length are set as $B = 10$MHz and $T = 0.2$s, respectively. It is assumed that the channel reciprocity holds for the downlink and uplink, and thus $g_1 = h_1$, $g_2 = h_2$. The channel power gain is modeled as $h_j = 10^{-3} d_j^{-\alpha} \phi_j$, $j \in \{1, 2, 12\}$, where ϕ_j represents the short-term fading which is assumed to be an exponentially distributed random variable with unit mean (Rayleigh fading). For distance d_j in meters (m) with the same path-loss

exponent α, a 30dB average signal power attenuation is assumed for all the channels at reference of 1m. We assume that $d_1 = 10$m, $d_2 = 6$m, $d_{12} = 6$m and $\alpha = 2$. The noise at the AP and UE_2 is assumed to have a white power of $N_0 = 10^{-9}$W. For each user UE_k, $k \in \{1, 2\}$, the CPU frequency f_k is uniformly selected from the set of $\{0.1, 0.2, \ldots, 1.0\}$GHz. We set $\nu_k = 0.8$ and $\kappa_k = 10^{-28}$, respectively. As for the computation tasks, the input data size and the required number of CPU cycles per bit follow the uniform distribution with $I_k \in [100, 500]$ Kbits and $C_k \in [1000, 2000]$ cycles/bit, respectively. The figures by simulations in the following subsections are based on 1000 independent realizations, in which h_k, f_k, I_k and C_k are randomly selected according to the above assumptions in each realization, modeling the real heterogeneous computing scenarios.

3.4.1 The Equivalence of Problem (P3.1) and Problem (P3.2)

In this subsection, we will verify the equivalence of problem (P3.1) and (P3.2) by simulations. The results of the average minimum transmit energy (AMTE) combining with the corresponding average minimum transmit power (AMTP) of the AP are shown in Figure 3.3 and Figure 3.4, versus the block length T and the same task-input data size $I = I_1 = I_2$, respectively.

From Figure 3.3, we can observe that the corresponding curves of AMTE and AMTP illustrate the same trend and property, verifying the equivalence of these two criteria in problem (P3.1) and (P3.2). It is shown that the proposed UC-JOPT scheme obviously outperforms the baselines. Specifically, the curves of UC-JOPT are much lower than those of UC-ET, indicating the effectiveness of the optimization for time allocation. Besides, the AMTE and AMTP of UC-JOPT are even less than half of those for IUC, which further displays the significance of

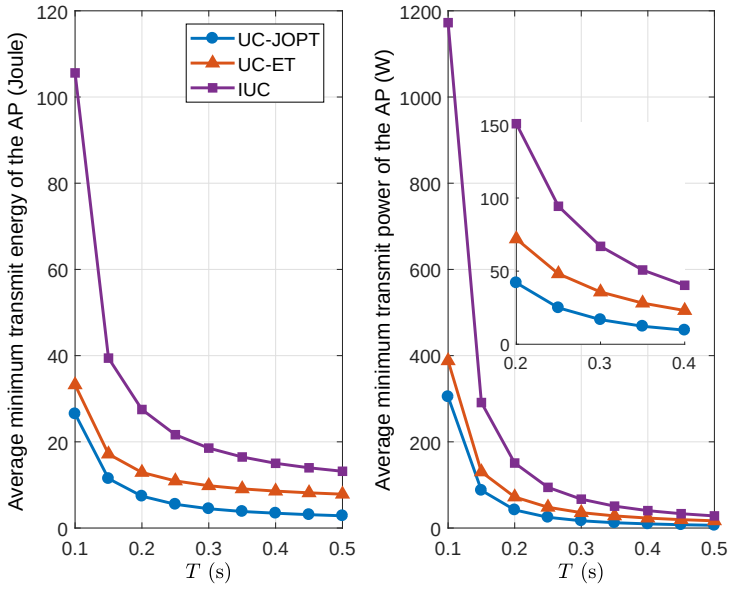


Figure 3.3: Average minimum transmit energy and power of the AP versus T.

user cooperation in handling the double-near-far effect in WPCNs. It is valuable to note that the gaps of AMTE (AMTP) between different schemes become more significant for a shorter block length (smaller T), demonstrating the superiority of the proposed UC-JOPT scheme in handling the latency-critical tasks.

Figure 3.4 also shows the equivalence between problem (P3.1) and (P3.2) by depicting both AMTE and AMTP versus the same task-input data size I. The AMTE and AMTP of all the schemes increase gradually with I, as expected. Besides, the performance improvement of the proposed UC-JOPT scheme is clearly displayed, and we can obtain similar results as those reported in Figure 3.3. Also, it is noted that the reduction of AMTE (AMTP) between different schemes become more obvious as I increases, which further indicates the advantage of the proposed UC-JOPT scheme in completing computation-intensive tasks.

The above results verify that the proposed UC-JOPT scheme is highly capable of dealing with computation-intensive latency-critical tasks and resisting

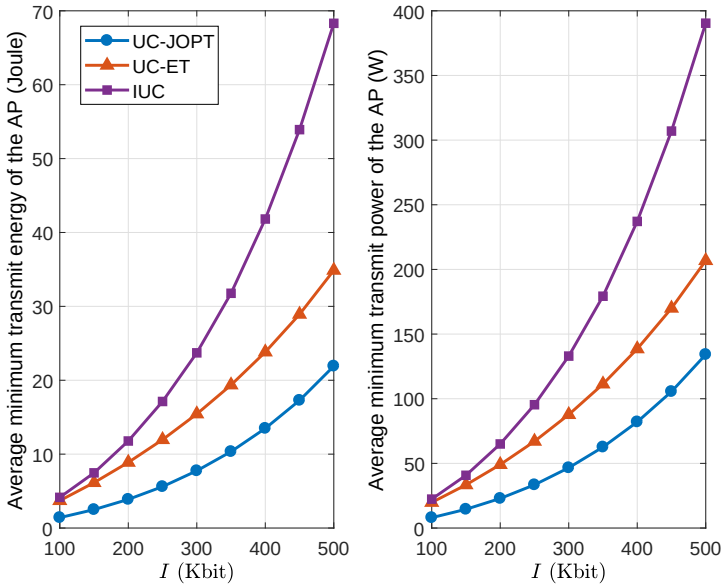


Figure 3.4: Average minimum transmit energy and power of the AP versus I.

the double-near-far effect in WPCNs by fully taking the benefits of joint-optimal resource allocation and user cooperation.

3.4.2 The Effects of Path Loss

From the expression of the channel power gain described above, it is understood that the path-loss exponent α and the distances d_1, d_2 and d_{12} have great influence on the value of h_1, h_2 and h_{12}, and thus further affect the AMTE (AMTP) of each scheme. In this part of simulations, we set same short-term fading parameters for UE_1 and UE_2, i.e., $\phi_1 = \phi_2$, and focus on the effect of α and distances on the AMTE. Setting $d_1 = 10\text{m}$, $d_2 = \xi d_1$, and $d_{12} = (1-\xi)d_1$, Figure 3.5 depicts the AMTE with respect to the distance ratio ξ for $\alpha = 1.5, 2, 2.5$.

From the results in Figure 3.5, we have the observation that the performance of the proposed UC-JOPT scheme is superior to the benchmarks, and the corresponding improvements are even more pronounced with a larger α, indicating that

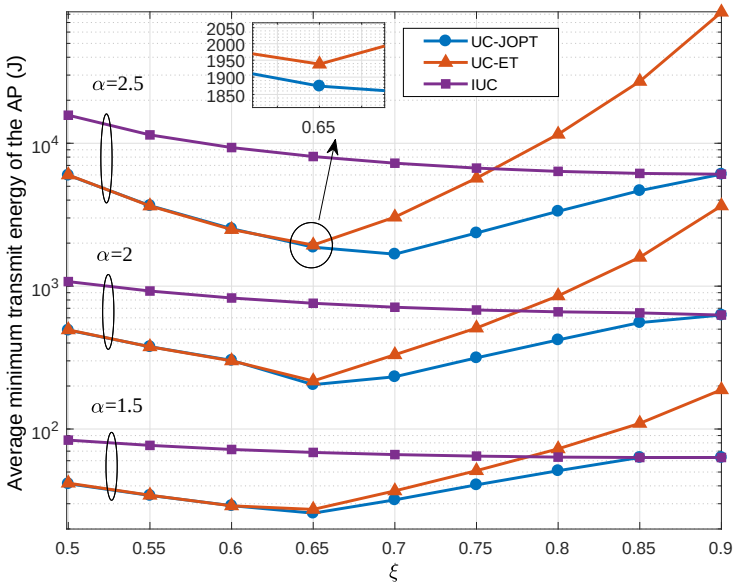


Figure 3.5: Average minimum transmit energy of the AP versus the distance ratio ξ.

the UC-JOPT scheme is highly effective in resisting the attenuation caused by path loss. It is also noticed that the AMTE curves of the two cooperative schemes, i.e., UC-JOPT and UC-ET, first decrease then increase with the distance ratio ξ, and there is a saddle point of ξ in each curve achieving the minimum AMTE. This is due to the fact that for the cooperative computation offloading schemes, the performance depends not only on h_2 but also h_{12}, and there exists a tradeoff between these two values. When ξ is small, the performance is limited by the value of h_{12}, and the AMTE curves decrease with ξ since h_{12} increases accordingly. Around the saddle point, the performance of both two cooperative schemes degrades with ξ as the decreasing h_2 plays a dominant role in this situation. This figure also shows that when ξ is less than the saddle point, the gaps between the two cooperative schemes are not that obvious, while the gaps widen obviously as ξ goes beyond the saddle point. It is interesting to note that the performance of the proposed UC-JOPT scheme converges to that of IUC as ξ gradually tends to 1 since both UE_1 and UE_2

suffer from severe signal attenuation, and $t_{21}^{\star}$ gradually approaches to 0. However, the performance of the UC-ET scheme is even worse than that of the IUC scheme when ξ becomes larger approaching to 1, which shows the importance and effect of optimizing the offloading time fraction.

3.5 Summary

In this chapter, we investigated the use of cooperative communications in computation offloading for a WPT-MEC system, in which an AP acts as an energy source via WPT and serves as an MEC server to assist two near-far UEs to complete their computation-intensive latency-critical tasks. Joint power and time allocation for cooperative computation offloading has been considered based on a block-based harvest-then-offload protocol, with the aim to minimize the total transmit energy of the AP for completing the computation tasks of the two UEs. A two-phase method was proposed to find the optimal solution of the offloading decisions and the AP's minimum-energy transmit power, where the joint power and time allocation are obtained in closed or semi-closed form with the given AP's energy transmit power. Numerical results not only revealed that the proposed scheme can achieve significant performance improvement but also demonstrated the effectiveness of the scheme in handling computation-intensive latency-critical tasks and resisting the double-near-far effect in WPCNs.

Chapter 4

Mobile Edge Computing in UAV-Assisted Relaying Systems

This chapter is based on our works published in [J2] and [C2] ([71] and [75]).

4.1 Introduction

Due to the attractive advantages of UAV for its easy deployment, flexible movement, and LoS connections, etc., it is a great attempt to leverage the technology of the UAV in MEC systems. Most existing MEC works concentrate either on the cellular-based MEC networks, where the UEs' tasks are completed by using the computing resources at the APs; or the UAV-enabled MEC architectures by exploiting the computing capabilities of the UAV processing servers. However, for the UEs with seriously degraded links to the AP due to severe blockage, it is impossible to take full use of the computing resources at the AP directly. Besides, due to the size-constrained resource-limited property of the UAVs, it is risky to rely only on the

UAVs to assist the UEs for completing their computation-intensive latency-critical tasks. For these reasons, we study a UAV-assisted MEC architecture in this chapter, where the computing resources at the UAV and the AP are utilized cooperatively at the same time to help the UEs complete their computation-intensive latency-critical tasks. In addition, the energy-efficient LoS transmissions of the UAV have been fully exploited since the UAV is not only served as a mobile computing server to help the UEs compute their tasks but also as a relay to further offload UEs' tasks to the AP for computing.

Our aim is to minimize the weighted sum energy consumption (WSEC) of the UAV and the UEs subject to the UEs' task constraints, the information-causality constraints, the bandwidth allocation constraints, and the UAV's trajectory constraints. The required optimization is nonconvex, and an alternating optimization algorithm is proposed to jointly optimize the computation resource scheduling, the bandwidth allocation, and the UAV's trajectory in an iterative fashion. Numerical results demonstrate that significant performance gain is obtained over conventional methods. Also, the advantages of the proposed algorithm are more prominent when handling computation-intensive latency-critical tasks.

The UAV in the considered scenario of this chapter acts as a MEC server as well as a relay, which is actually an aerial communication platform. It is interesting to note that the technology of user-cooperation can also be applied in the UAV scenarios especially when the UAVs are acting as aerial users, where the ground users and UAV users can cooperatively help each other to complete the computation tasks. For example, the users (current-strong users) with more idle radio/computing resources can share these resources with the users (current-weak users) with insufficient radio/computing resources due to the currently high

computing demand such as operating computation-intensive applications. The cooperation can be either computing the current-weak users' offloaded data with shared computing resources or relaying the offloaded data to the AP with shared radio resources. The incentive behind this kind of user cooperation could be that the current-strong users sharing their resources to other current-weak users can enjoy the shared resource if they become current-weak users in the situation with insufficient resources for completing the intensive computing workloads in the future. The work considering this UAV-user cooperation strategy will be considered as one of our future works.

4.2 System Model and Problem Formulation

As shown in Figure 4.1, a UAV-assisted MEC system is considered, which consists of an AP, a cellular-connected UAV, and K ground UEs, all being equipped with a single antenna. The UAV and UEs are all assumed to have an on-board communication circuit and on-board computing processor powered by their embedded battery, while the AP is capable of providing high-speed transmission rate with grid power supply and is endowed with an ultra-high performance processing server. It is also assumed that each UE has a bit-wise-independent computation-intensive task, and the UAV acts as an assistant to help the UEs complete their computation tasks by providing both MEC and relaying services. For providing MEC service, the UAV shares its computing resources with the UEs to help compute their tasks; while for the relaying service, the UAV forwards part of the UEs' offloaded tasks to the AP for computing with the purpose of satisfying the latency constraints or saving its own energy.

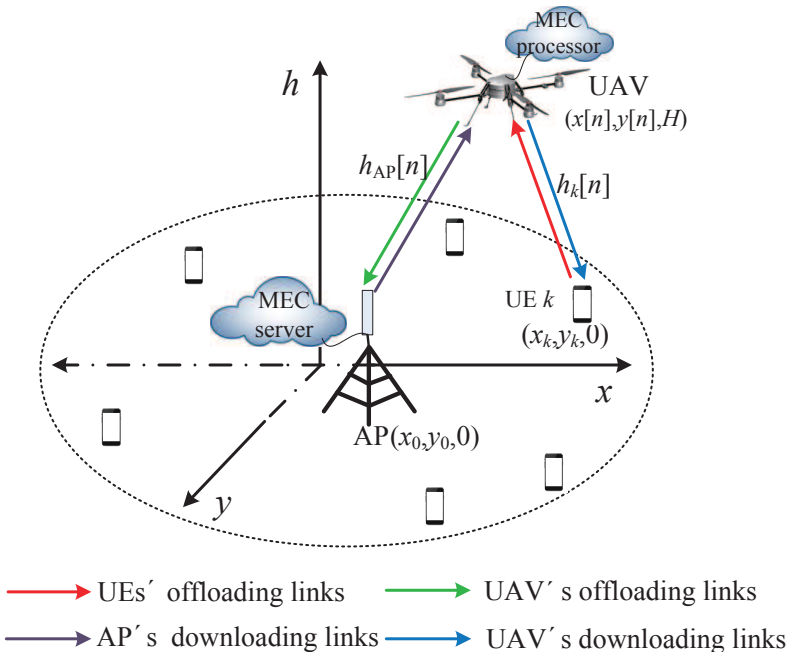


Figure 4.1: An illustration of the UAV-assisted MEC architecture, where the UAV serves as an MEC server to help the ground UEs compute their offloaded tasks as well as a possible relay to further forward the offloaded tasks to the AP with more powerful computing resources.

4.2.1 Channel Model and Coordinate System

A 3D Euclidean coordinate system is adopted, whose coordinates are measured in meters (m). We assume that the locations of the AP and all the UEs are fixed on the ground with zero altitude, with the location of the AP being $\widetilde{\mathbf{s}}_0 = (x_0, y_0, 0)$. Let $\mathcal{K} = \{1, \ldots, K\}$ denote the set of the UEs, with $\widetilde{\mathbf{s}}_k = (x_k, y_k, 0)$ representing the location of UE $k \in \mathcal{K}$. It is assumed that the locations of UEs are known to the UAV for designing its trajectory [59]. We assume that the UAV flies at a fixed altitude $H > 0$ during the task completion time T, which corresponds to the minimum altitude that is appropriate to the work terrain and can avoid buildings without the requirement of frequent descending and ascending.

For ease of exposition, the finite task completion time T in seconds (s) is discretized into N equal time slots each with a duration of $\tau = T/N$, where τ

is sufficiently small such that the UAV's location can be assumed to be unchanged during each slot. The initial and final horizontal locations of the UAV are preset as $\mathbf{u}_{\mathrm{I}} = (x_{\mathrm{I}}, y_{\mathrm{I}})$ and $\mathbf{u}_{\mathrm{F}} = (x_{\mathrm{F}}, y_{\mathrm{F}})$, respectively. Let $\mathcal{N} = \{1, \ldots, N\}$ denote the set of the N time slots. At the n-th time slot, the UAV's horizontal location is denoted as $\mathbf{u}[n] \equiv \mathbf{u}(n\tau) = (x[n], y[n])$ with $\mathbf{u}[0] = \mathbf{u}_{\mathrm{I}}$ and $\mathbf{u}[N] = \mathbf{u}_{\mathrm{F}}$. It is assumed that the UAV flies with a constant speed in each time slot, denoted as $v[n]$, which should satisfy the following maximum speed constraint

$$v[n] = \frac{\|\mathbf{u}[n] - \mathbf{u}[n-1]\|}{\tau} \leq V_{\max},\ n \in \mathcal{N}, \tag{4.1}$$

where $V_{\max}$ is the predetermined maximum speed of the UAV, and $V_{\max} \geq \|\mathbf{u}_{\mathrm{F}} - \mathbf{u}_{\mathrm{I}}\|/T$ establishes to make sure that at least one feasible trajectory of the UAV exists.

Similar to [59], the wireless channels between the UAV and the AP as well as the UEs are assumed to be dominated by LoS links, which is verified by recent field experiments done by Qualcomm [125].[1] Thus, the channel power gain between the UAV and the AP and between the UAV and UE k at the time slot n can be, respectively, given by

$$h_{\mathrm{AP}}[n] = h_0 d_{\mathrm{AP}}^{-2} = \frac{h_0}{\|\mathbf{u}[n] - \mathbf{s}_0\|^2 + H^2},\ n \in \mathcal{N}, \tag{4.2}$$

$$h_k[n] = h_0 d_k^{-2} = \frac{h_0}{\|\mathbf{u}[n] - \mathbf{s}_k\|^2 + H^2},\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.3}$$

where h_0 is the channel power gain at a reference distance of $d_0 = 1$m; d_{AP} and d_k are respectively the distances between the UAV and the AP as well as the UE k at the

[1] It is of great value to extend our work on the probabilistic LoS and Rician fading channel models when we consider the scenarios where the UAV's flying altitude changes according to the work terrain.

n-th time slot with $\mathbf{s}_0 = (x_0, y_0)$ and $\mathbf{s}_k = (x_k, y_k)$ denoting the horizontal locations of the AP and UE k, $k \in \mathcal{K}$. It is assumed that the channel reciprocity establishes in our considered scenario, and thus the offloading and downloading channels between the UEs and the UAV are identical. In this chapter, the direct links between UEs and the AP are assumed to be negligible due to e.g., severe blockage,[2] which means that the UEs cannot directly offload their task-input bits to the AP unless with the assistance of the UAV. The motivation behind this scenario is based on the fact that it is more important to guarantee the UEs' computation tasks being completed within the given limited time T with as little UEs' energy as possible, than dropping their tasks or letting the UEs compute their takes locally at the cost of exhausting their energy.

4.2.2 Computation Task Model and Execution Methods

The computation task of UE $k \in \mathcal{K}$ is denoted as a positive tuple $[I_k, C_k, O_k, T_k]$, where I_k denotes the size (in bits) of the computation task-input data (e.g., the program codes and input parameters), C_k is the amount of required computing resource for computing 1-bit of input data (i.e., the number of CPU cycles required), $O_k \in (0, 1)$ is the ratio of task-output data size to that of the task-input data, i.e., the output data size should be $O_k I_k$ for UE k, and T_k is the maximum tolerable latency with $T_k \leq T, k \in \mathcal{K}$. In this chapter, we only consider the case that $T_k = T$ for all $k \in \mathcal{K}$. It should be noted that the UEs' task-input bits are bit-wise independent and can be arbitrarily divided to facilitate parallel trade-offs between local computing at the UEs and computation offloading to the UAV or further to the AP with the assistance of the UAV. In other words, the UEs can accomplish their computation

[2]The general case with direct links between the UEs and the AP is a promising extension of our current work.

tasks in a partial offloading fashion [14] with the following three ways.

4.2.2.1 Local Computing at UEs

Each UE can perform local computing and computation offloading simultaneously since local computing at the UEs does not need radio resources such as bandwidth. To efficiently use the energy for local computing, the UEs leverage the DVFS technique, and thus the energy consumed for local computing can be adaptively controlled by adjusting the UEs' CPU frequency during each time slot [20]. The CPU frequency of UE k during time slot n is denoted as $f_k[n]$ (cycles/second). Thus, the computation bits and energy consumption of UE k during time slot n for local computing can be, respectively, expressed as

$$L_k^{\mathrm{loc}}[n] = \tau f_k[n]/C_k,\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.4}$$

$$E_k^{\mathrm{loc}}[n] = \tau \kappa_k f_k^3[n],\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.5}$$

where κ_k is the effective capacitance coefficient of UE k that depends on its processor's chip architecture.

4.2.2.2 Task Offloaded to the UAV for Computing

The UEs' remaining task-input data should be computed remotely, first by offloading to the UAV, and then one part of the data being computed at the UAV while the other part further offloaded to the AP for computing. In order to avoid interference among the UEs during the offloading process, we adopt the TDMA protocol. Each slot is further divided into K equal durations $\delta = T/(NK)$, and UE k offloads its task-input data in the k-th duration. Let $L_k^{\mathrm{off}}[n]$ denote the offloaded bits of UE k in its allocated duration at time slot n, and thus the corresponding energy consumption

of UE k at slot n for computation offloading can be calculated as

$$E_k^{\text{off}}[n] = \delta P_k^{\text{off}}[n] \equiv \frac{\delta N_0}{h_k[n]} \left(2^{\frac{L_k^{\text{off}}[n]}{\delta B_k^{\text{off}}[n]}} - 1 \right),\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.6}$$

where $P_k^{\text{off}}[n]$ is the transmit power of UE k for offloading $L_k^{\text{off}}[n]$ computation bits to the UAV at time slot n, $B_k^{\text{off}}[n]$ is the corresponding allocated bandwidth for UE k, and N_0 denotes the noise power at the UAV.[3]

Assume that the UAV also adopts the DVFS technique to improve its energy efficiency for computing, and its adjustable CPU frequency in the k-th duration of slot n for computing UE k's offloaded task is denoted as $f_{\text{U},k}[n]$. Hence, the completed computation bits and the energy consumption of the UAV for computing UE k's task at time slot n can be, respectively, given by

$$L_{\text{U},k}[n] = \delta f_{\text{U},k}[n]/C_k,\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.7}$$

$$E_{\text{U},k}[n] = \delta \kappa_{\text{U}} f_{\text{U},k}^3[n],\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.8}$$

where κ_{U} is the effective capacitance coefficient of the UAV. Note that computing $L_{\text{U},k}[n]$ bits of UE k's task-input data will produce $O_k L_{\text{U},k}[n]$ bits of task-output data, which should be downloaded from the UAV to the UE k later.

4.2.2.3 Task Offloaded to the AP for Computing

Part of the UEs' offloaded task-input data at the UAV will be offloaded to the AP's processing server for computing. To better distinguish the offloading signals from different UEs, the TDMA protocol with K equal time divisions ($\delta = T/(NK)$) is also adopted in this case. Let $L_{\text{U},k}^{\text{off}}[n]$ denote the number of UE k's task-

[3]Without loss of generality, we assume that the noise power at any node in the system is considered the same as N_0 in this chapter.

input bits being offloaded from the UAV to the AP at time slot n. Thus, the corresponding energy consumption of the UAV for offloading UE k's task at slot n can be calculated as

$$E_{\mathrm{U},k}^{\mathrm{off}}[n] = \delta P_{\mathrm{U},k}^{\mathrm{off}}[n] \equiv \frac{\delta N_0}{h_{\mathrm{AP}}[n]} \left(2^{\frac{L_{\mathrm{U},k}^{\mathrm{off}}[n]}{\delta B_{\mathrm{U},k}^{\mathrm{off}}[n]}} - 1 \right),\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.9}$$

where $P_{\mathrm{U},k}^{\mathrm{off}}[n]$ and $B_{\mathrm{U},k}^{\mathrm{off}}[n]$ are respectively the transmit power and the allocated bandwidth of the UAV for offloading UE k's task to the AP at time slot n. After computing the $L_{\mathrm{U},k}^{\mathrm{off}}[n]$ input bits at the AP, $O_k L_{\mathrm{U},k}^{\mathrm{off}}[n]$ bits of computation results for UE k will be generated. As the AP is integrated with an ultra-high-performance processing server, the computing time is negligible. The AP will send the computation results back to the UAV in the TDMA manner using a separate bandwidth. Since the AP is supplied with grid power and can support ultra-high transmission rate, the download transmission time from the AP to the UAV is also assumed negligible.[4]

For the latter two offloading methods, the generated computation results at the UAV (including the results from UAV's computing and received from the AP) will then be downloaded back to the corresponding UEs. It is assumed that the UAV is equipped with a data buffer with sufficiently large size, and it is capable of storing each UE's offloaded data and the corresponding computation results separately. Besides, we assume that the UAV operates in a frequency-division duplex (FDD) mode in each UE's operation duration δ with separate bandwidths allocated for task reception from UEs ($\{B_k^{\mathrm{off}}[n]\}$), task offloading transmission to the AP ($\{B_{\mathrm{U},k}^{\mathrm{off}}[n]\}$),

[4]Once the AP receives the forwarded $L_{\mathrm{U},k}^{\mathrm{off}}[n]$ bits input data from the UAV in the k-th duration of the n-th time slot, it will immediately decode, compute the data, and then send the induced $O_k L_{\mathrm{U},k}^{\mathrm{off}}[n]$ bits of output data back to the UAV, all with ultra-low latency that is negligible compared with the length of each duration δ, which means that the UAV can receive the task-output data from the AP in the same duration of its offloading process.

and task results downloading transmission to the UEs ($\{B_{\mathrm{U},k}^{\mathrm{down}}[n]\}$), with a total bandwidth B satisfying the constraint

$$B_k^{\mathrm{off}}[n] + B_{\mathrm{U},k}^{\mathrm{off}}[n] + B_{\mathrm{U},k}^{\mathrm{down}}[n] = B,\ k \in \mathcal{K}, n \in \mathcal{N}. \tag{4.10}$$

The UEs' computation results are subsequently transmitted by the UAV using TDMA similar to the UEs' offloading process, each with an equal duration δ in each time slot. Let $L_{\mathrm{U},k}^{\mathrm{down}}[n]$ denote the bits of task-output data being downloaded from the UAV to UE k at time slot n. Hence, the corresponding energy consumption of the UAV can be calculated as

$$E_{\mathrm{U},k}^{\mathrm{down}}[n] = \delta P_{\mathrm{U},k}^{\mathrm{down}}[n] \equiv \frac{\delta N_0}{h_k[n]}\left(2^{\frac{L_{\mathrm{U},k}^{\mathrm{down}}[n]}{\delta B_{\mathrm{U},k}^{\mathrm{down}}[n]}} - 1\right),\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.11}$$

where $P_{\mathrm{U},k}^{\mathrm{down}}[n]$ is the transmit power of the UAV for downloading UE k's task-output data at time slot n.

Note that at each time slot n, the UAV can only compute or forward the task-input data that has already been received from the UEs. By assuming that the processing delay, e.g., the delay for decoding and computing preparation, at the UAV is one time slot, then we have the following information-causality constraint:

$$\sum_{i=2}^{n}\left(\frac{\delta f_{\mathrm{U},k}[i]}{C_k} + L_{\mathrm{U},k}^{\mathrm{off}}[i]\right) \le \sum_{i=1}^{n-1} L_k^{\mathrm{off}}[i], \tag{4.12}$$

for $n \in \mathcal{N}_2$ and $k \in \mathcal{K}$ where $\mathcal{N}_2 = \{2, \ldots, N-1\}$. Similarly, at each time slot n, the UAV can only transmit the task-output data corresponding to the task-input data that has already been computed at the UAV or offloaded for computing at the AP

and received the results. Thus, we have another information-causality constraint:

$$\sum_{i=3}^{n} L_{\mathrm{U},k}^{\mathrm{down}}[i] \leq O_k \sum_{i=2}^{n-1} \left(\frac{\delta f_{\mathrm{U},k}[i]}{C_k} + L_{\mathrm{U},k}^{\mathrm{off}}[i] \right), \tag{4.13}$$

for $n \in \mathcal{N}_3$ and $k \in \mathcal{K}$ where $\mathcal{N}_3 = \{3, \ldots, N\}$. It is clear that the UEs should not offload at the last two slots, while the UAV should not compute or forward the received input data of UEs at the first and the last slots as well as not transmit the output data to the UEs in the first two slots. Hence, we have $L_k^{\mathrm{off}}[N-1] = L_k^{\mathrm{off}}[N] = 0$, $f_{\mathrm{U},k}[1] = f_{\mathrm{U},k}[N] = 0$, $L_{\mathrm{U},k}^{\mathrm{off}}[1] = L_{\mathrm{U},k}^{\mathrm{off}}[N] = 0$, and $L_{\mathrm{U},k}^{\mathrm{down}}[1] = L_{\mathrm{U},k}^{\mathrm{down}}[2] = 0$.

4.2.3 Problem Formulation

Considering the fact that the traditional battery-based UEs and UAVs are usually power-limited, one major problem that the UAV-assisted MEC system faces will be energy. Hence, in this chapter, we try to minimize the WSEC of the UAV as well as all the UEs during the whole task completion time T. In the previous subsection, we have obtained the energy consumption of the UEs and the UAV for task offloading/downloading and computation. In fact, the energy consumption for UAV's propulsion is also considerable which is greatly affected by the UAV's trajectory, and hence should be taken into account. With the assumption that the time slot duration τ is sufficiently small, the UAV's flying during each slot can be regarded as straight-and-level flight with constant speed $v[n]$. Taking a fixed-wing UAV as an example [58, 59], its propulsion energy consumption at time slot n can be expressed as

$$E_{\mathrm{U}}^{\mathrm{prob}}[n] = \tau \left(\theta_1 v^3[n] + \frac{\theta_2}{v[n]} \right), \; n \in \mathcal{N}, \tag{4.14}$$

where θ_1 and θ_2 are two parameters related to the UAV's weight, wing area, wing span efficiency, and air density, etc. Combining with the above analysis, we obtain the total energy consumption of UE k and the UAV in each time slot n as

$$E_k[n] = E_k^{\mathrm{loc}}[n] + E_k^{\mathrm{off}}[n],\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.15}$$

$$E_{\mathrm{U}}[n] = \sum_{k=1}^{K} \left(E_{\mathrm{U},k}[n] + E_{\mathrm{U},k}^{\mathrm{off}}[n] + E_{\mathrm{U},k}^{\mathrm{down}}[n] \right) + E_{\mathrm{U}}^{\mathrm{prob}}[n],\ n \in \mathcal{N}. \tag{4.16}$$

In our considered scenario, the UEs' CPU computing frequencies $\{f_k[n]\}$, their offloading task-input bits $\{L_k^{\mathrm{off}}[n]\}$ and the corresponding allocated bandwidth $\{B_k^{\mathrm{off}}[n]\}$; the UAV's CPU computing frequencies $\{f_{\mathrm{U},k}[n]\}$, its forwarding (further offloading) task-input bits $\{L_{\mathrm{U},k}^{\mathrm{off}}[n]\}$ and downloading task-output bits $\{L_{\mathrm{U},k}^{\mathrm{down}}[n]\}$ as well as the corresponding allocated bandwidths $\{B_{\mathrm{U},k}^{\mathrm{off}}[n]\}$, $\{B_{\mathrm{U},k}^{\mathrm{down}}[n]\}$ for different UEs; along with the UAV's trajectory, i.e., $\{\mathbf{u}[n]\}$, will be jointly optimized to minimize the WSEC. To this end, the WSEC minimization problem can be formulated as problem (P4.1) given below

$$(\mathrm{P4.1}):\ \min_{\mathbf{z},\mathbf{B},\mathbf{u}}\ \sum_{n=1}^{N} \left(w_{\mathrm{U}} E_{\mathrm{U}}[n] + \sum_{k=1}^{K} w_k E_k[n] \right) \tag{4.17a}$$

$$\text{s.t.}\ \sum_{i=2}^{n} \left(\frac{\delta f_{\mathrm{U},k}[i]}{C_k} + L_{\mathrm{U},k}^{\mathrm{off}}[i] \right) \le \sum_{i=1}^{n-1} L_k^{\mathrm{off}}[i],\ \forall n \in \mathcal{N}_2, k \in \mathcal{K}, \tag{4.17b}$$

$$\sum_{i=3}^{n} L_{\mathrm{U},k}^{\mathrm{down}}[i] \le O_k \sum_{i=2}^{n-1} \left(\frac{\delta f_{\mathrm{U},k}[i]}{C_k} + L_{\mathrm{U},k}^{\mathrm{off}}[i] \right), \forall n \in \mathcal{N}_3, k \in \mathcal{K}, \tag{4.17c}$$

$$\sum_{n=2}^{N-1} \left(\frac{\delta f_{\mathrm{U},k}[n]}{C_k} + L_{\mathrm{U},k}^{\mathrm{off}}[n] \right) = \sum_{n=1}^{N-2} L_k^{\mathrm{off}}[n],\ \forall k \in \mathcal{K}, \tag{4.17d}$$

$$\sum_{n=3}^{N} L_{\mathrm{U},k}^{\mathrm{down}}[n] = O_k \sum_{n=2}^{N-1} \left(\frac{\delta f_{\mathrm{U},k}[n]}{C_k} + L_{\mathrm{U},k}^{\mathrm{off}}[n] \right),\ \forall k \in \mathcal{K}, \tag{4.17e}$$

$$\sum_{n=1}^{N} \frac{\tau}{C_k} f_k[n] + \sum_{n=1}^{N-2} L_k^{\mathrm{off}}[n] = I_k,\ \forall k \in \mathcal{K}, \tag{4.17f}$$

$$B_k^{\text{off}}[n] + B_{\text{U},k}^{\text{off}}[n] + B_{\text{U},k}^{\text{down}}[n] = B,\ \forall n \in \mathcal{N}, k \in \mathcal{K}, \tag{4.17g}$$

$$f_k[n] \geq 0,\ \forall n \in \mathcal{N}, k \in \mathcal{K}, \tag{4.17h}$$

$$L_k^{\text{off}}[N-1] = L_k^{\text{off}}[N] = 0,\ L_k^{\text{off}}[n] \geq 0,\ \forall n \in \mathcal{N}_1, k \in \mathcal{K}, \tag{4.17i}$$

$$f_{\text{U},k}[1] = f_{\text{U},k}[N] = 0,\ f_{\text{U},k}[n] \geq 0,\ \forall n \in \mathcal{N}_2, k \in \mathcal{K}, \tag{4.17j}$$

$$L_{\text{U},k}^{\text{off}}[1] = L_{\text{U},k}^{\text{off}}[N] = 0,\ L_{\text{U},k}^{\text{off}}[n] \geq 0,\ \forall n \in \mathcal{N}_2, k \in \mathcal{K}, \tag{4.17k}$$

$$L_{\text{U},k}^{\text{down}}[1] = L_{\text{U},k}^{\text{down}}[2] = 0,\ L_{\text{U},k}^{\text{down}}[n] \geq 0,\ \forall n \in \mathcal{N}_3, k \in \mathcal{K}, \tag{4.17l}$$

$$B_k^{\text{off}}[N-1] = B_k^{\text{off}}[N] = 0,\ B_k^{\text{off}}[n] \geq 0,\ \forall n \in \mathcal{N}_1, k \in \mathcal{K}, \tag{4.17m}$$

$$B_{\text{U},k}^{\text{off}}[1] = B_{\text{U},k}^{\text{off}}[N] = 0,\ B_{\text{U},k}^{\text{off}}[n] \geq 0,\ \forall n \in \mathcal{N}_2, k \in \mathcal{K}, \tag{4.17n}$$

$$B_{\text{U},k}^{\text{down}}[1] = B_{\text{U},k}^{\text{down}}[2] = 0,\ B_{\text{U},k}^{\text{down}}[n] \geq 0, \forall n \in \mathcal{N}_3, k \in \mathcal{K}, \tag{4.17o}$$

$$\mathbf{u}[0] = \mathbf{u}_{\text{I}},\ \mathbf{u}[N] = \mathbf{u}_{\text{F}}, \tag{4.17p}$$

$$\|\mathbf{u}[n] - \mathbf{u}[n-1]\| \leq V_{\max}\tau,\ \forall n \in \mathcal{N}, \tag{4.17q}$$

where we have $\mathbf{z} \triangleq \{\mathbf{z}_k[n]\}_{k\in\mathcal{K},n\in\mathcal{N}}$ and $\mathbf{B} \triangleq \{\mathbf{B}_k[n]\}_{k\in\mathcal{K},n\in\mathcal{N}}$ with $\mathbf{z}_k[n] \triangleq \{f_k[n], L_k^{\text{off}}[n], f_{\text{U},k}[n], L_{\text{U},k}^{\text{off}}[n], L_{\text{U},k}^{\text{down}}[n]\}$ and $\mathbf{B}_k[n] \triangleq \{B_k^{\text{off}}[n], B_{\text{U},k}^{\text{off}}[n], B_{\text{U},k}^{\text{down}}[n]\}$, respectively, denote the sets of the computational resource scheduling variables and the bandwidth allocation variables for UE k in time slot n, $\mathbf{u} \triangleq \{\mathbf{u}[n]\}_{n\in\mathcal{N}}$ denotes the set of the UAV's horizontal locations for all the slots, i.e., the trajectory of the UAV, and $\mathcal{N}_1 = \{1, \ldots, N-2\}$. In problem (P4.1), (4.17a) is the objective function for minimizing the WSEC where w_{U} and $\{w_k\}_{k\in\mathcal{K}}$ represent the weights of the UAV and UEs, respectively, which trade-offs the energy consumption between the UAV and UEs, and consider the priority/fairness among the UEs. Also, (4.17b) and (4.17c) are the two information-causality constraints, while (4.17d)–(4.17f) are the UEs' computation task constraints to make sure that all the UEs' computation task-input data has been

computed and the corresponding task-output data has been received. The bandwidth constraints are in (4.17g), while (4.17h)–(4.17o) ensure the non-negativeness of the optimization variables. (4.17p) and (4.17q) specify the UAV's initial and final horizontal locations, and its maximum speed constraints.

4.3 Proposed Three-Step Alternating Optimization Algorithm

The formulated WSEC minimization problem (P4.1) is a complicated non-convex optimization problem because of the non-convex objective function where non-linear couplings exist among the variables $L_k^{\text{off}}[n]$ and $B_k^{\text{off}}[n]$, $L_{\text{U},k}^{\text{off}}[n]$ and $B_{\text{U},k}^{\text{off}}[n]$, $L_{\text{U},k}^{\text{down}}[n]$ and $B_{\text{U},k}^{\text{down}}[n]$ for $k \in \mathcal{K}, n \in \mathcal{N}$, and these variables are also strongly coupled with the trajectory of the UAV, i.e., $\mathbf{u}[n]$. To address these issues, we propose a three-step alternating optimization algorithm to solve the problem. In the first step, the computation resource scheduling variables in $\mathbf{z}$ are optimized by solving the problem with given UAV's trajectory $\mathbf{u}$ and bandwidth allocation $\mathbf{B}$; and then in the second step, the bandwidth allocation variables in $\mathbf{B}$ will be optimized with the same given UAV's trajectory $\mathbf{u}$ and the optimized $\mathbf{z}$ obtained in the first step; and finally in the third step, we focus on designing the UAV's trajectory $\mathbf{u}$ with the optimized variables $\mathbf{z}$ and $\mathbf{B}$.

We assume that the proposed three-step alternating optimization algorithm for solving the formulated WSEC minimization problem (P4.1) is carried out at the servers of the AP, and then the the UAV and UEs will be informed with the obtained solution sent from the AP as the control information. Hence, the UEs and the UAV can perform the offloading and computing operations based

on the computing resource scheduling, bandwidth allocation and UAV's trajectory contained in the obtained solution. This strategy is applicable in practice since the considered UEs and the UAV are cellular-based, and the control information is sent periodically. Considering the fact the grid-powered AP is with super computing capability and high transmit power, the cost of time and energy for implementing the propose algorithm and sending the solution back to the UEs and the UAV should be acceptable in general. The details for the three-step algorithm are presented in the next section.

4.3.1 Computation Resource Scheduling with Fixed UAV's Trajectory and Bandwidth Allocation

A subproblem of (P4.1) is the computation resource scheduling problem (P4.1.1), where the UAV's trajectory $\mathbf{u}$ and bandwidth allocation $\mathbf{B}$ are given as fixed. In this case, the time-dependent channels $\{h_{\mathrm{AP}}[n]\}_{n\in\mathcal{N}}$ and $\{h_k[n]\}_{k\in\mathcal{K},n\in\mathcal{N}}$ defined in (4.2) and (4.3) are also known. Besides, the non-linear couplings among the offloading/downloading task-input/task-output bits ($L_k^{\mathrm{off}}[n], L_{\mathrm{U},k}^{\mathrm{off}}[n], L_{\mathrm{U},k}^{\mathrm{down}}[n]$) with their corresponding allocated bandwidths ($B_k^{\mathrm{off}}[n], B_{\mathrm{U},k}^{\mathrm{off}}[n], B_{\mathrm{U},k}^{\mathrm{down}}[n]$) no longer exist. The resource scheduling problem (P4.1.1) is convex with a convex objective function and convex constraints, which is expressed as

$$(\mathrm{P4.1.1}): \min_{\mathbf{z}} \ \sum_{n=1}^{N}\left(w_{\mathrm{U}}E_{\mathrm{U}}^{(1)}[n]+\sum_{k=1}^{K}w_kE_k[n]\right) \tag{4.18a}$$

$$\text{s.t.} \ \ (4.17\mathrm{b})-(4.17\mathrm{f}),\ (4.17\mathrm{h})-(4.17\mathrm{l}), \tag{4.18b}$$

where $E_{\mathrm{U}}^{(1)}[n]=\sum_{k=1}^{K}\left(E_{\mathrm{U},k}[n]+E_{\mathrm{U},k}^{\mathrm{off}}[n]+E_{\mathrm{U},k}^{\mathrm{down}}[n]\right)$. In order to gain more insights into the solution, we leverage the Lagrange method [123] to solve problem (P4.1.1),

and the optimal solution of problem (P4.1.1) is given in the following theorem.

Theorem 4.1. *The optimal solution of problem (P4.1.1) related to UE $k \in \mathcal{K}$ is given below*

$$f_k^*[n] = \sqrt{\frac{[\beta_k^*]^+}{3C_k w_k \kappa_k}},\ n \in \mathcal{N}, \tag{4.19}$$

$$L_k^{\text{off}*}[n] = \begin{cases} \delta B_k^{\text{off}}[n]\left[\varphi_k^{\text{off}}[n] + \log_2\left[\sum\limits_{i=n+1}^{N-1} \lambda_{k,i}^* + \beta_k^* - \eta_k^*\right]^+\right]^+, & n \in \mathcal{N}_1, \\ 0, \quad n = N-1 \text{ or } N, \end{cases} \tag{4.20}$$

$$f_{\text{U},k}^*[n] = \begin{cases} \sqrt{\dfrac{\left[\eta_k^* - O_k\rho_k^* + O_k\sum\limits_{i=n+1}^{N}\mu_{k,i}^* - \sum\limits_{i=n}^{N-1}\lambda_{k,i}^*\right]^+}{3C_k w_{\text{U}}\kappa_{\text{U}}}}, & n \in \mathcal{N}_2, \\ 0, \quad n = 1 \text{ or } N, \end{cases} \tag{4.21}$$

$$L_{\text{U},k}^{\text{off}*}[n] = \begin{cases} \delta B_{\text{U},k}^{\text{off}}[n]\left[\varphi_{\text{U},k}^{\text{off}}[n] + \log_2\left[\eta_k^* - O_k\rho_k^* + O_k\sum\limits_{i=n+1}^{N}\mu_{k,i}^* - \sum\limits_{i=n}^{N-1}\lambda_{k,i}^*\right]^+\right]^+, & n \in \mathcal{N}_2, \\ 0, \quad n = 1 \text{ or } N, \end{cases} \tag{4.22}$$

$$L_{\text{U},k}^{\text{down}*}[n] = \begin{cases} \delta B_{\text{U},k}^{\text{down}}[n]\left[\varphi_{\text{U},k}^{\text{down}}[n] + \log_2\left[\rho_k^* - \sum\limits_{i=n}^{N}\mu_{k,i}^*\right]^+\right]^+, & n \in \mathcal{N}_3, \\ 0, \quad n = 1 \text{ or } 2, \end{cases} \tag{4.23}$$

where

$$\varphi_k^{\text{off}}[n] = \log_2\frac{B_k^{\text{off}}[n]h_k[n]}{w_k N_0 \ln 2},\ n \in \mathcal{N}_1, \tag{4.24}$$

$$\varphi_{\mathrm{U},k}^{\mathrm{off}}[n] = \log_2 \frac{B_{\mathrm{U},k}^{\mathrm{off}}[n] h_{\mathrm{AP}}[n]}{w_{\mathrm{U}} N_0 \ln 2},\ n \in \mathcal{N}_2, \tag{4.25}$$

$$\varphi_{\mathrm{U},k}^{\mathrm{down}}[n] = \log_2 \frac{B_{\mathrm{U},k}^{\mathrm{down}}[n] h_k[n]}{w_{\mathrm{U}} N_0 \ln 2},\ n \in \mathcal{N}_3, \tag{4.26}$$

are denoted as the offloading/downloading priority indicators for the UEs in each given slot. Also, $\lambda_{k,n}^* \geq 0$ *and* $\mu_{k,n}^* \geq 0$ *for* $k \in \mathcal{K}, n \in \mathcal{N}$ *are respectively the optimal Lagrange multipliers (dual variables) associated with the inequality constraints* (4.17b) *and* (4.17c) *in problem (P4.1.1) (or P4.1), while* η_k^*, ρ_k^* *and* β_k^* *are respectively the optimal Lagrange multipliers associated with the equality constraints* (4.17d)–(4.17f) *for* $k \in \mathcal{K}$.

Proof. See Appendix B.1. □

Remark 4.1. *(Intuitive Explanation). From the expressions relating to the computation resource scheduling parameters in* ***Theorem*** *4.1, we observe that* $\{L_k^{\mathrm{off}}[n]\}$, $\{L_{\mathrm{U},k}^{\mathrm{off}}[n]\}$, *and* $\{L_{\mathrm{U},k}^{\mathrm{down}}[n]\}$ *are monotonically increasing with* $\{\varphi_k^{\mathrm{off}}[n]\}$, $\{\varphi_{\mathrm{U},k}^{\mathrm{off}}[n]\}$ *and* $\{\varphi_{\mathrm{U},k}^{\mathrm{down}}[n]\}$ *when they are positive. It coincides with the intuition that more input (or output) data should be offloaded (or downloaded) with larger* $\{\varphi_k^{\mathrm{off}}[n]\}$, $\{\varphi_{\mathrm{U},k}^{\mathrm{off}}[n]\}$ *and* $\{\varphi_{\mathrm{U},k}^{\mathrm{down}}[n]\}$, *corresponding to the scenarios with larger bandwidths, channel power gains and smaller weights for energy consumption.*

Remark 4.2. *(Decreasing Offloading and Increasing Downloading Data Size).* ***Theorem*** *4.1 sheds light on the fact that* $L_k^{\mathrm{off}*}[n]$ *decreases with the time slot index* n *while* $L_{\mathrm{U},k}^{\mathrm{down}*}[n]$ *increases with* n *for the reason that* $\sum_{i=n+1}^{N-1} \lambda_{k,i}^*$ *and* $\sum_{i=n}^{N} \mu_{k,i}^*$ *in* (4.20) *and* (4.23) *decrease with* n *as* $\lambda_{k,i}^* \geq 0$ *and* $\mu_{k,i}^* \geq 0$. *This indicates that the resource allocated for UEs' task offloading gradually decreases while that for UAV's downloading gradually increases as time goes by.*

It is necessary to obtain the optimal values of the Lagrange multipliers, i.e.,

$\boldsymbol{\lambda}^* = \{\lambda^*_{k,n}\}_{k\in\mathcal{K},n\in\mathcal{N}}$, $\boldsymbol{\mu}^* = \{\mu^*_{k,n}\}_{k\in\mathcal{K},n\in\mathcal{N}}$, $\boldsymbol{\eta}^* = \{\eta^*_k\}_{k\in\mathcal{K}}$, $\boldsymbol{\rho}^* = \{\rho^*_k\}_{k\in\mathcal{K}}$ and $\boldsymbol{\beta}^* = \{\beta^*_k\}_{k\in\mathcal{K}}$ since they play important roles in determining the optimal computation resource scheduling $\mathbf{z}^*$ according to **Theorem** 4.1. In this chapter, we adopt a subgradient-based algorithm to obtain the optimal dual variables in $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$ related to the inequality constraints (4.17b), (4.17c), as described in the following **Lemma** 4.1.

Lemma 4.1. *The dual variables $\{\lambda_{k,n}\}$ and $\{\mu_{k,n}\}$ obtained at the $(j+1)$-th ($j = 1, 2, \dots$) iteration of the subgradient-based algorithm are expressed as*

$$\lambda_{k,n,j+1} = [\lambda_{k,n,j} - \varepsilon_j^{(\lambda)} \Delta\lambda_{k,n,j}]^+,\ k \in \mathcal{K}, n \in \mathcal{N}_2, \tag{4.27}$$

$$\mu_{k,n,j+1} = [\mu_{k,n,j} - \varepsilon_j^{(\mu)} \Delta\mu_{k,n,j}]^+,\ k \in \mathcal{K}, n \in \mathcal{N}_3, \tag{4.28}$$

with the corresponding subgradients given as

$$\Delta\lambda_{k,n,j} = \sum_{i=1}^{n-1} L^{\text{off}*}_{k,j}[i] - \sum_{i=2}^{n} \left(\frac{\delta f^*_{\text{U},k,j}[i]}{C_k} + L^{\text{off}*}_{\text{U},k,j}[i] \right), \tag{4.29}$$

$$\Delta\mu_{k,n,j} = O_k \sum_{i=2}^{n-1} \left(\frac{\delta f^*_{\text{U},k,j}[i]}{C_k} + L^{\text{off}*}_{\text{U},k,j}[i] \right) - \sum_{i=3}^{n} L^{\text{down}*}_{\text{U},k,j}[i], \tag{4.30}$$

*where $\varepsilon_j^{(\lambda)}$ and $\varepsilon_j^{(\mu)}$ respectively denote the iterative steps for obtaining the dual variables in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ at the j-th iteration [126]. Also, $\{L^{\text{off}*}_{k,j}[n]\}$, $\{f^*_{\text{U},k,j}[n]\}$, $\{L^{\text{off}*}_{\text{U},k,j}[n]\}$, $\{L^{\text{down}*}_{\text{U},k,j}[n]\}$ are the computation resource scheduling variables obtained through **Theorem** 4.1 with the dual variables obtained at the j-th iteration, i.e., $\boldsymbol{\lambda}_j = \{\lambda_{k,n,j}\}_{k\in\mathcal{K},n\in\mathcal{N}}$, $\boldsymbol{\mu}_j = \{\mu_{k,n,j}\}_{k\in\mathcal{K},n\in\mathcal{N}}$, $\boldsymbol{\eta}_j = \{\eta_{k,j}\}_{k\in\mathcal{K}}$, $\boldsymbol{\rho}_j = \{\rho_{k,j}\}_{k\in\mathcal{K}}$ and $\boldsymbol{\beta}_j = \{\beta_{k,j}\}_{k\in\mathcal{K}}$.*

Proof. See Appendix B.2. □

Besides, the bi-section search method is used to obtain the optimal dual variables in $\boldsymbol{\eta}^*$, $\boldsymbol{\rho}^*$ and $\boldsymbol{\beta}^*$ related to the equality constraints (4.17d)–(4.17f), as summarized in the following **Lemma** 4.2.

Lemma 4.2. *With the obtained $\boldsymbol{\lambda}_{j+1}$ and $\boldsymbol{\mu}_{j+1}$ above, the corresponding $\boldsymbol{\eta}_{j+1}$, $\boldsymbol{\rho}_{j+1}$ and $\boldsymbol{\beta}_{j+1}$ can be obtained by bi-section search of $\{\beta_{k,j+1}\}_{k\in\mathcal{K}} \in [0, \{\beta_{k,\max}\}_{k\in\mathcal{K}})$ where $\beta_{k,\max} = 3C_k w_k \kappa_k (\frac{I_k C_k}{T})^2$. For each given $\beta_{k,j+1} \in [0, \beta_{k,\max})$, the corresponding $\eta_{k,j+1}$ and $\rho_{k,j+1}$ can be obtained with another two bi-section searches within $\eta_{k,j+1} \in [\eta_{k,j+1}^{\text{low}}, \eta_{k,j+1}^{\text{up}}]$ and $\rho_{k,j+1} \in [\rho_{k,j+1}^{\text{low}}, \rho_{k,j+1}^{\text{up}}]$ to make the expressions satisfy* (B.3.1)=(B.3.2) *and* (B.3.1)=(B.3.3)*, respectively, in Appendix B.3, where the expressions of $\eta_{k,j+1}^{\text{low}}$, $\eta_{k,j+1}^{\text{up}}$, $\rho_{k,j+1}^{\text{low}}$, and $\rho_{k,j+1}^{\text{up}}$ are given in* (B.3.5)–(B.3.8) *in Appendix B.3. The optimal $\beta_{k,j+1}$, $\eta_{k,j+1}$ and $\rho_{k,j+1}$ should satisfy* (B.3.1)=(B.3.4)*.*

Proof. See Appendix B.3. □

The optimal dual variables $\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$ and $\boldsymbol{\eta}^*, \boldsymbol{\rho}^*, \boldsymbol{\beta}^*$ can be finally obtained when the subgradient algorithm converges [127], and the bi-section searches terminate. The results in [127] implies that the subgradient method is guaranteed to converge to the optimal value of a convex optimization problem if the diminishing step size rule is adopted. Even if the constant step length are employed, the subgradient method is capable of finding a ϵ-suboptimal point within a finite number of steps. It is known that the Lagrange dual problem is convex no matter the original problem is convex or not, and thus the convergence of the subgradient algorithm for solving the dual problem to obtain the dual variables can be guaranteed.

Hence, the corresponding convergence of the Lagrange method in combination with the subgradient algorithm and bi-section search method for obtaining the

optimal solutions of problem (P4.1.1) in (4.18) can be further guaranteed according to [123, 127], based on the fact that the problem (P4.1.1) is proved to be convex.

4.3.2 Bandwidth Allocation with Fixed UAV's Trajectory and Computation Resource Scheduling

Here, another subproblem of (P4.1), denoted as the bandwidth allocation problem (P4.1.2) is considered to optimize $\mathbf{B}$ with the same given UAV's trajectory $\mathbf{u}$ and the optimized computation resource scheduling parameters in $\mathbf{z}$. The bandwidth allocation problem (P4.1.2) is expressed as

$$(\mathrm{P4.1.2}): \min_{\mathbf{B}} \ \sum_{n=1}^{N}\left(w_{\mathrm{U}}E_{\mathrm{U}}^{(2)}[n]+\sum_{k=1}^{K}w_k E_k^{\mathrm{off}}[n]\right) \tag{4.31a}$$

$$\text{s.t.} \ \ (4.17\mathrm{g}),\ (4.17\mathrm{m})-(4.17\mathrm{o}), \tag{4.31b}$$

where $E_{\mathrm{U}}^{(2)}[n]=\sum\limits_{k=1}^{K}\left(E_{\mathrm{U},k}^{\mathrm{off}}[n]+E_{\mathrm{U},k}^{\mathrm{down}}[n]\right)$. It can be easily proved that problem (P4.1.2) is convex with convex objective function and constraints. To gain more insights into the structure of the optimal solution, we again leverage the Lagrange method [123] to solve this problem, and the optimal solution to problem (P4.1.2) is given in the following theorem.

Theorem 4.2. *The optimal solution of problem (P4.1.2) related to UE $k \in \mathcal{K}$ is given by*

$$B_k^{\mathrm{off}*}[n]=\begin{cases}\dfrac{\frac{\ln 2}{2}L_k^{\mathrm{off}}[n]}{\delta W_0\left[\frac{\ln 2}{2}\left(\frac{\phi_{k,n}}{w_k}h_k[n]L_k^{\mathrm{off}}[n]\right)^{\frac{1}{2}}\right]}, & n\in\mathcal{N}_1,\\ 0, \quad n=N-1 \text{ or } N,\end{cases} \tag{4.32}$$

$$B_{\mathrm{U},k}^{\mathrm{off}*}[n]=\begin{cases}\dfrac{\frac{\ln 2}{2}L_{\mathrm{U},k}^{\mathrm{off}}[n]}{\delta W_0\left[\frac{\ln 2}{2}\left(\frac{\phi_{k,n}}{w_{\mathrm{U}}}h_{\mathrm{AP}}[n]L_{\mathrm{U},k}^{\mathrm{off}}[n]\right)^{\frac{1}{2}}\right]}, & n\in\mathcal{N}_2,\\ 0, \quad n=1 \text{ or } N,\end{cases} \tag{4.33}$$

$$B_{\mathrm{U},k}^{\mathrm{down}*}[n]=\begin{cases}\dfrac{\frac{\ln 2}{2}L_{\mathrm{U},k}^{\mathrm{down}}[n]}{\delta W_0\left[\frac{\ln 2}{2}\left(\frac{\phi_{k,n}}{w_{\mathrm{U}}}h_{k}[n]L_{\mathrm{U},k}^{\mathrm{down}}[n]\right)^{\frac{1}{2}}\right]}, & n\in\mathcal{N}_3,\\ 0, \quad n=1 \text{ or } 2,\end{cases} \tag{4.34}$$

where $\phi_{k,n}=\frac{\nu_{k,n}^*}{\delta^2 N_0\ln 2}$ *with* $\{\nu_{k,n}^*\}_{k\in\mathcal{K},n\in\mathcal{N}}$ *being the optimal Lagrange multipliers (dual variables) associated with the equality constraints in* (4.17g) *of problem (P4.1.2) (or P4.1), and* $W_0(x)$ *is the principal branch of the Lambert* W *function defined as the solution of* $W_0(x)e^{W_0(x)}=x$ *[124].*

Proof. See Appendix B.4. □

Lemma 4.3. *(Exclusive Bandwidth Allocation). According to the optimal bandwidth allocation results in* ***Theorem*** *4.2 combining with the equality constraints in* (4.17g)*, we have*

$$B_k^{\mathrm{off}*}[n]=B, \text{ if } L_k^{\mathrm{off}}[n]>0, L_{\mathrm{U},k}^{\mathrm{off}}[n]=L_{\mathrm{U},k}^{\mathrm{down}}[n]=0, \tag{4.35}$$

$$B_{\mathrm{U},k}^{\mathrm{off}*}[n]=B, \text{ if } L_{\mathrm{U},k}^{\mathrm{off}}[n]>0, L_k^{\mathrm{off}}[n]=L_{\mathrm{U},k}^{\mathrm{down}}[n]=0, \tag{4.36}$$

$$B_{\mathrm{U},k}^{\mathrm{down}*}[n]=B, \text{ if } L_{\mathrm{U},k}^{\mathrm{down}}[n]>0, L_k^{\mathrm{off}}[n]=L_{\mathrm{U},k}^{\mathrm{off}}[n]=0, \tag{4.37}$$

where the whole bandwidth is exclusively occupied when only one of $L_k^{\mathrm{off}}[n]$*,* $L_{\mathrm{U},k}^{\mathrm{off}}[n]$*,* $L_{\mathrm{U},k}^{\mathrm{down}}[n]$ *is positive for any* $k\in\mathcal{K}$*,* $n\in\mathcal{N}$*. Also, it is always sure that*

$$B_k^{\mathrm{off}*}[1]=B,\ B_{\mathrm{U},k}^{\mathrm{down}*}[N]=B,\ k\in\mathcal{K}. \tag{4.38}$$

The optimal Lagrange multipliers $\{\nu_{k,n}^*\}$ for obtaining the optimal bandwidth

allocation in **Theorem** 4.2 correspond to $\{\phi_{k,n}\}$, which should make the equality constraints in (4.17g) satisfied. In fact, $\phi_{k,n}$ can be obtained effectively with the bi-section search when the bandwidth is not exclusively occupied, i.e., at least two of $L_k^{\text{off}}[n]$, $L_{\text{U},k}^{\text{off}}[n]$, $L_{\text{U},k}^{\text{down}}[n]$ are positive, since $\{B_k^{\text{off}*}[n]\}_{n\in\mathcal{N}_1}$, $\{B_{\text{U},k}^{\text{off}*}[n]\}_{n\in\mathcal{N}_2}$ and $\{B_{\text{U},k}^{\text{down}*}[n]\}_{n\in\mathcal{N}_3}$ are all monotonically decreasing functions with respect to (w.r.t.) $\{\phi_{k,n}\}$ according to the property of the W_0 function. Besides, we can obtain tight search ranges using the results in **Lemma** 4.4.

Lemma 4.4. *A tight bi-section search range of $\phi_{k,n}$ ($k \in \mathcal{K}$) for any slot $n \in \mathcal{N}$ with non-exclusive bandwidth allocation is given as $\phi_{k,n} \in [\phi_{k,n}^{\min}, \phi_{k,n}^{\max}]$ where*

$$\phi_{k,n}^{\min} \text{ (or } \phi_{k,n}^{\max}) = \min \text{ (or } \max) \tag{4.39}$$
$$\begin{cases} \{\phi_{\text{UE},k,n}^{\text{off}}(B/3), \phi_{\text{U},k,n}^{\text{off}}(B/3), \phi_{\text{U},k,n}^{\text{down}}(B/3)\}, & \textbf{case 1} \\ \{\phi_{\text{UE},k,n}^{\text{off}}(B/2), \phi_{\text{U},k,n}^{\text{off}}(B/2)\}, & \textbf{case 2} \\ \{\phi_{\text{UE},k,n}^{\text{off}}(B/2), \phi_{\text{U},k,n}^{\text{down}}(B/2)\}, & \textbf{case 3} \\ \{\phi_{\text{U},k,n}^{\text{off}}(B/2), \phi_{\text{U},k,n}^{\text{down}}(B/2)\}, & \textbf{case 4} \end{cases}$$

where **case 1**-**case 4** *are distinguished by the values of $L_k^{\text{off}}[n]$, $L_{\text{U},k}^{\text{off}}[n]$ and $L_{\text{U},k}^{\text{down}}[n]$ for each $n \in \mathcal{N}$. For* **case 1***, all the three parameters have positive values; for* **case 2***, $L_{\text{U},k}^{\text{down}}[n] = 0$; for* **case 3***, $L_{\text{U},k}^{\text{off}}[n] = 0$; for* **case 4***, $L_k^{\text{off}}[n] = 0$. In* (4.39)*,*

$$\phi_{\text{UE},k,n}^{\text{off}}(x) = \frac{w_k L_k^{\text{off}}[n]}{\delta^2 x^2 h_k[n]} e^{\frac{L_k^{\text{off}}[n]\ln 2}{\delta x}},\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.40}$$

$$\phi_{\text{U},k,n}^{\text{off}}(x) = \frac{w_{\text{U}} L_{\text{U},k}^{\text{off}}[n]}{\delta^2 x^2 h_{\text{AP}}[n]} e^{\frac{L_{\text{U},k}^{\text{off}}[n]\ln 2}{\delta x}},\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.41}$$

$$\phi_{\text{U},k,n}^{\text{down}}(x) = \frac{w_{\text{U}} L_{\text{U},k}^{\text{down}}[n]}{\delta^2 x^2 h_k[n]} e^{\frac{L_{\text{U},k}^{\text{down}}[n]\ln 2}{\delta x}},\ k \in \mathcal{K}, n \in \mathcal{N}, \tag{4.42}$$

which are the value of $\phi_{k,n}$ obtained by letting the expressions of $B_k^{\text{off}*}[n]$, $B_{\text{U},k}^{\text{off}*}[n]$

and $B_{\mathrm{U},k}^{\mathrm{down}*}[n]$ *in* (4.32)–(4.34) *equal to* x.

4.3.3 UAV Trajectory Design With Fixed Computation Resource Scheduling and Bandwidth Allocation

Here, the subproblem for designing the UAV's trajectory $\mathbf{u}$ is considered, which we refer to it as the UAV trajectory design problem (P4.1.3), by assuming that the computation resource scheduling $\mathbf{z}$ and bandwidth allocation $\mathbf{B}$ are given as fixed with the previously optimized values. Hence, the UAV trajectory design problem (P4.1.3) can be rewritten as

$$(\mathrm{P4.1.3}): \min_{\mathbf{u}} \quad \sum_{n=1}^{N}\left(w_{\mathrm{U}}E_{\mathrm{U}}^{(3)}[n]+\sum_{k=1}^{K}w_k E_k^{\mathrm{off}}[n]\right) \tag{4.43a}$$

$$\text{s.t.} \quad \text{(4.17p)},\ \text{(4.17q)}, \tag{4.43b}$$

where $E_{\mathrm{U}}^{(3)}[n]=E_{\mathrm{U}}^{\mathrm{prob}}[n]+\sum\limits_{k=1}^{K}\left(E_{\mathrm{U},k}^{\mathrm{off}}[n]+E_{\mathrm{U},k}^{\mathrm{down}}[n]\right)$. It is noted that the $E_{\mathrm{U}}^{\mathrm{prob}}[n]$ defined in (4.14) with $v[n]$ in (4.1) is not a convex function of $\mathbf{u}$. In order to address this issue, we first define an upper bound of $E_{\mathrm{U}}^{\mathrm{prob}}[n]$ as follows

$$\widetilde{E}_{\mathrm{U}}^{\mathrm{prob}}[n]=\tau\left(\theta_1 v^3[n]+\frac{\theta_2}{\widetilde{v}[n]}\right),\ n\in\mathcal{N}, \tag{4.44}$$

by introducing a variable $\widetilde{v}[n]$ and a constraint $v[n]\geq\widetilde{v}[n]$, which is equivalent to $\|\mathbf{u}[n]-\mathbf{u}[n-1]\|^2\geq\widetilde{v}^2[n]\tau^2$. This constraint is still non-convex, and we then leverage the successive convex approximation (SCA) technique to solve this issue. The left-hand side of the constraint is convex versus $\mathbf{u}$ and can be approximated as its linear lower bound by using the first-order Taylor expansion at a local point $\mathbf{u}_i$, where $i=1,2,\ldots$ denotes the iteration index of the SCA method. Hence, the

additional constraint can be approximated as a convex one as follows

$$\widetilde{v}^2[n]\tau^2 - 2(\mathbf{u}_i[n] - \mathbf{u}_i[n-1])^T(\mathbf{u}[n] - \mathbf{u}[n-1]) \\ + \|\mathbf{u}_i[n] - \mathbf{u}_i[n-1]\|^2 \leq 0,\ n \in \mathcal{N}. \tag{4.45}$$

The approximated problem of (P4.1.3) with $\{\widetilde{E}_{\mathrm{U}}^{\mathrm{prob}}[n]\}$, $\{\widetilde{v}[n]\}$ and the additional constraint (4.45) is convex w.r.t. $\mathbf{u}$ and $\{\widetilde{v}[n]\}$. However, the UAV's locations in different slots are coupled with each other as in (4.17q), and thus it is difficult to obtain a closed-form solution of $\mathbf{u}$. In this case, we resort to the software CVX [128] to solve the approximated problem of (P4.1.3).

4.3.4 Algorithm, Convergence, and Complexity

Based on the aforementioned analysis of the alternating optimization for the computation resource scheduling $\mathbf{z}$, the bandwidth allocation $\mathbf{B}$ and the UAV's trajectory $\mathbf{u}$ in each subproblem, **Algorithm** 4.1 is proposed to solve the original problem (P4.1) for obtaining the solution $\{\mathbf{z}^*, \mathbf{B}^*, \mathbf{u}^*\}$.[5]

The convergence of **Algorithm** 4.1 is easy to prove in light of the guaranteed convergence of the loop Repeat 1.1 in Step 1, the bi-section search in Step 2 and the CVX solving process based on the SCA method in Step 3 [123]. The lower-bounded objective function of problem (P4.1) will monotonically decrease with the iteration index ζ by optimizing $\mathbf{z}$, $\mathbf{B}$ and $\mathbf{u}$ alternatingly in each subproblem, which further guarantees the convergence of the algorithm.

In addition, **Algorithm** 4.1 is easy to implement and the corresponding complexity is acceptable. In Step 1, the complexity mainly comes from the subgradient

[5]The proposed method is not theoretically optimal due to problem non-convexity, but its performance gain is verified by the simulation results.

Algorithm 4.1 Three-Step Algorithm for Solving Problem (P4.1)

1: **Set** B, T, N, K, h_0, N_0, H, $V_{\max}$, θ_1, θ_2, $\mathbf{u}_{\mathrm{I}}$, $\mathbf{u}_{\mathrm{F}}$, w_{U}, κ_{U}, $\mathbf{s}_0$, $\{\mathbf{s}_k, w_k, I_k, C_k, O_k, \kappa_k\}_{k\in\mathcal{K}}$, two tolerant thresholds ϵ_1 and ϵ, and the iterative steps $\{\varepsilon_j^{(\lambda)}\}$ and $\{\varepsilon_j^{(\mu)}\}$;

2: **Initialize** the iteration index $\zeta = 1$ and $\mathbf{u}_1$, $\mathbf{B}_1$;

3: **Repeat 1**

4: **Initialize** $j = 1$, as well as $\boldsymbol{\lambda}_1$, $\boldsymbol{\mu}_1$;

5: **Step 1: Repeat 1.1**

6: a) Obtain $\boldsymbol{\eta}_j$, $\boldsymbol{\rho}_j$, $\boldsymbol{\beta}_j$ with $\boldsymbol{\lambda}_j$, $\boldsymbol{\mu}_j$ through **Lemma** 4.2;

b) Obtain $\mathbf{z}_{\zeta,j}^* = \left\{\{f_{k,j}^*[n]\}, \{L_{k,j}^{\mathrm{off}*}[n]\}, \{f_{\mathrm{U},k,j}^*[n]\}, \{L_{\mathrm{U},k,j}^{\mathrm{off}*}[n]\}, \{L_{\mathrm{U},k,j}^{\mathrm{down}*}[n]\}\right\}$ through **Theorem** 4.1 with $\boldsymbol{\lambda}_j$, $\boldsymbol{\mu}_j$, $\boldsymbol{\eta}_j$, $\boldsymbol{\rho}_j$, $\boldsymbol{\beta}_j$ and $\mathbf{u}_\zeta$, $\mathbf{B}_\zeta$;

c) Calculate the WSEC $E_j^{(1)}$ by substituting $\mathbf{z}_{\zeta,j}^*$, $\mathbf{B}_\zeta$, $\mathbf{u}_\zeta$ into the objective function of problem (P4.1.1);

d) $j = j + 1$;

e) Update $\boldsymbol{\lambda}_j$ and $\boldsymbol{\mu}_j$ according to **Lemma** 4.1;

7: **End Repeat 1.1** until convergence, i.e., $|E_j^{(1)} - E_{j-1}^{(1)}| < \epsilon_1$ $(j > 1)$, and obtain optimal $\mathbf{z}_{\zeta+1} = \mathbf{z}_{\zeta,j}^*$;

8: **Step 2: Bi-section search** of $\{\nu_{k,n}\}$ to find the optimal $\{\nu_{k,n}^*\}$ and obtain the $\mathbf{B}_{\zeta+1} = \mathbf{B}_\zeta^* = \left\{\{B_k^{\mathrm{off}*}[n]\}, \{B_{\mathrm{U},k}^{\mathrm{off}*}[n]\}, \{B_{\mathrm{U},k}^{\mathrm{down}*}[n]\}\right\}$ according to **Theorem** 4.2, **Lemma** 4.3 and **Lemma** 4.4 with given $\mathbf{u}_\zeta$ and $\mathbf{z}_{\zeta+1}$;

9: **Step 3:** Solve the approximated problem of (P4.1.3) by CVX based on the SCA method, so as to obtain the optimal solution $\mathbf{u}_{\zeta+1}$ with the given $\mathbf{z}_{\zeta+1}$, $\mathbf{B}_{\zeta+1}$;

10: $\zeta = \zeta + 1$;

11: Calculate the WSEC E_ζ, by substituting $\mathbf{z}_\zeta$, $\mathbf{B}_\zeta$, and $\mathbf{u}_\zeta$ into the objective function of problem (P4.1);

12: **End Repeat 1** until convergence, i.e., $|E_\zeta - E_{\zeta-1}| < \epsilon$ $(\zeta > 2)$, and obtain the minimum WSEC E_ζ with the solution $\mathbf{z}^* = \mathbf{z}_\zeta$, $\mathbf{B}^* = \mathbf{B}_\zeta$, $\mathbf{u}^* = \mathbf{u}_\zeta$;

method for obtaining $\{\lambda_{k,n}\}$, $\{\mu_{k,n}\}$, and the bi-section searches of $\{\beta_k\}$, $\{\rho_k\}$ and $\{\eta_k\}$ in each iteration of Repeat 1.1. Let $\varepsilon_{\rm sub} > 0$, and $\varepsilon_\beta, \varepsilon_\rho, \varepsilon_\eta > 0$ denote the computational accuracies of the subgradient method and the bi-section searches for $\{\beta_k\}$, $\{\rho_k\}$ and $\{\eta_k\}$. Thus, the corresponding complexity can be calculated as $\mathcal{O}(1/\varepsilon_{\rm sub}^2 + K\log_2(1/\varepsilon_\beta)(\log_2(1/\varepsilon_\rho) + \log_2(1/\varepsilon_\eta)))$. In Step 2, the complexity is from the bi-section search of $\{\nu_{k,n}\}$, which is calculated as $\mathcal{O}(KN\log_2(1/\varepsilon_\nu))$, where ε_ν is the corresponding computational accuracy. In Step 3, the complexity mainly focuses on solving the approximation problem of (P4.1.3) by CVX, which is acceptable in general.

4.4 Numerical Results

In this section, simulation results are presented to evaluate the performance of the proposed algorithm against the benchmarking schemes. The effects of the key parameters will be analyzed, including the relative location of the AP ($\mathbf{s}_0$),[6] the computation task sizes of UEs (I_k for $k \in \mathcal{K}$), the task completion time for UEs (T), the size ratio of task-output data to task-input data (O_k for $k \in \mathcal{K}$), the weight for energy consumption of the UAV ($w_{\rm U}$), and the iteration index of the alternating optimization algorithm (ζ). The basic simulation parameters are listed in **Table** 4.1 unless specified otherwise.

4.4.1 Trajectory of the UAV

In this subsection, numerical results for the trajectory of the UAV are given to shed light on the effects of the task sizes of UEs ($[I_1, I_2, I_3, I_4]$) and the relative location

[6]In order to properly show the effects of the relative location of the AP to UEs on UAV's trajectory and the performance, we fix the locations of the UEs and vary the location of AP even though AP is usually fixed in practice.

Table 4.1: Simulation Parameters

Parameter	**Symbol**	**Value**
The total system bandwidth	B	30 MHz
The total task completion time	T	10 seconds
Number of time slots	N	50
Number of ground UEs	K	4
The channel power gain at a reference distance of d_0=1 m	h_0	-30dB
The noise power	N_0	-60dBm
The fixed altitude of the UAV	H	10 m
The maximum available speed of the UAV	$V_{\max}$	10 m/s
The UAV's propulsion energy consumption related parameters	(θ_1, θ_2)	(0.00614,15.976)
The initial and final horizontal location of the UAV	$\mathbf{u}_{\mathrm{I}}$, $\mathbf{u}_{\mathrm{F}}$	$(-5,-5)$, $(5,-5)$
The horizontal locations of the UEs	$\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4$	$(5,5)$, $(-5,5)$, $(-5,-5)$, $(5,-5)$
The effective switched capacitance of the UAV and UEs	$\kappa_{\mathrm{U}}, \kappa_k (k \in \mathcal{K})$	10^{-28}
The weight for energy consumption of the UAV	w_{U}	0.2
The weight for energy consumption of the UEs	w_k $(k \in \mathcal{K})$	1
Required CPU cycles per bit	C_k $(k \in \mathcal{K})$	1000 cycles/bit
UEs' task-input data size	I_k $(k \in \mathcal{K})$	400 Mbits
UEs' task size ratio of output data to input data	O_k $(k \in \mathcal{K})$	0.8
The tolerant thresholds	ϵ_1 and ϵ	10^{-4}

of the AP ($\mathbf{s}_0$). In Figure 4.2, the UAV's flying trajectories are depicted in different scenarios. It should be noted that the total task size of UEs is same for the cases in (a), (c), (d) and (f), i.e., 1400 Mbits, while the cases for (b) and (e) are with larger total task size, e.g., 1800 Mbits. From these results in Figure 4.2, we can observe that the trajectory of the UAV is heavily reliant on the relative location of the AP and the distribution of UEs' task sizes.

For the scenario of $\mathbf{s}_0 = (0, 0)$, the AP is surrounded by the UEs and at the center of the UEs' distributed area. We can observe that the UAV tends to fly close to the UEs with large task sizes and tries to be not too far away from the AP when the total task sizes of UEs are moderate as the results in cases (a) and (c). When the total task size becomes larger and the distribution of UEs' task sizes becomes more

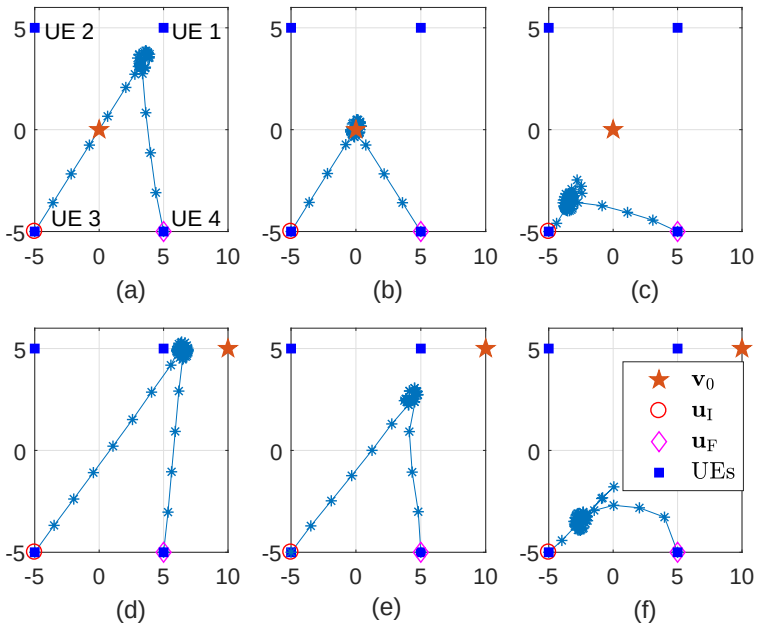


Figure 4.2: The trajectories of the UAV in the situations with different horizontal location of the AP and task size allocation of the UEs: $\mathbf{s}_0 = (0, 0)$ for (a), (b) and (c), $\mathbf{s}_0 = (10, 5)$ for (d), (e) and (f); $[I_1, I_2, I_3, I_4] = [6, 2, 4, 2] \times 10^2$Mbits for (a) and (d), $[I_1, I_2, I_3, I_4] = [6, 4, 6, 2] \times 10^2$Mbits for (b) and (e), $[I_1, I_2, I_3, I_4] = [2, 2, 6, 4] \times 10^2$Mbits for (c) and (f).

average, the UAV tends to fly close to the AP as the result in case (b). These three cases indicate that for the scenario where the AP is located at the center of UEs' distributed area, the distribution of the UEs' task sizes plays an important role in the UAV's trajectory. In addition, the effect of the AP's location will become more dominant when the UEs' total task size becomes larger, which coincides with the intuition that more task-input data will be offloaded to the AP in this situation so as to reduce the WSEC by making use of the super computing resources at the AP.

For the scenario of $\mathbf{s}_0 = (10, 5)$, the AP is located outside the distributed area of the UEs and its average distance to the UEs is relatively larger than the above scenario. In this situation, the effects of AP's location on the UAV's trajectories are more prominent, where the comparison between the cases (a) and (d), (b) and (e), (c) and (f) can properly explain this.

The reason behind these results in Figure 4.2 is that there exists a tradeoff between the distribution of UEs' task sizes and the relative location of the AP to the UEs. In other words, getting closer to the UEs with larger task sizes can reduce the UEs' offloading and the UAV's downloading energy consumption, while being closer to the AP will reduce the UAV's offloading energy consumption, and thus the UAV has to find a balance between these two factors meanwhile taking its own flying energy consumption into consideration, so as to minimize the WSEC of the UAV and UEs through optimizing its flying trajectory.

4.4.2 Performance Improvement

Here, we focus on the performance improvement of the proposed algorithm. The performance of the baselines is also provided for comparison, including the "Direct Trajectory" scheme where the UAV flies from its initial location to the final location directly with an average speed; the "Offloading Only" scheme where the UEs just rely on task offloading to the UAV and the AP for computing without local computing by the UEs themselves; the "Equal Bandwidth" scheme indicating the solution that the whole bandwidth is equally divided by the active $B_k^{\text{off}}[n]$, $B_{\text{U},k}^{\text{off}}[n]$, and $B_{\text{U},k}^{\text{down}}[n]$, for $n \in \mathcal{N}$ and $k \in \mathcal{K}$ without bandwidth optimization; and the "Local Computing" scheme, where the UEs rely on their own computing resources to complete their computation tasks without offloading. Note that the former four schemes are all offloading schemes. To better illustrate the effects of the AP's relative location on the performance, we present all the results in two scenarios given in Figure 4.2, i.e., $\mathbf{s}_0 = (0, 0)$ and $\mathbf{s}_0 = (10, 5)$.

Figure 4.3 shows the WSEC results versus the uniform task size $I = I_k$ for $k \in \mathcal{K}$. All the curves in the figures increase with I as expected since more energy

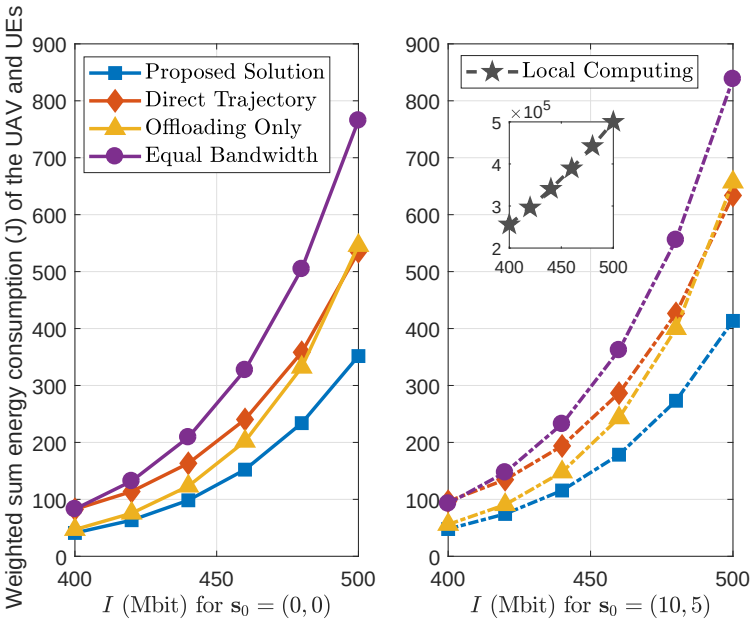


Figure 4.3: The WSEC of the UAV and UEs versus the uniform task size: $I = I_k$ for $k \in \mathcal{K}$.

will be consumed by completing tasks with more input data. It can be seen that great performance improvement can be achieved by leveraging the proposed solution in comparison with all the baseline schemes in both scenarios. It is clear that the performance of the "Local Computing" scheme is far worse than the other schemes with computation offloading, verifying the importance of edge computing through offloading. Specifically, the WSECs of the "Proposed Solution" are almost one thousandth of that for the "Local Computing" scheme, presenting the tremendous benefits the UEs obtained by deploying the UAV as an assistant for computing and relaying. In addition, the WSECs of the "Proposed Solution" are half less than those of the "Equal Bandwidth" scheme and they are almost quarter less than those of the "Direct Trajectory" scheme. The "Offloading Only" scheme performs well with relatively small task sizes, e.g., $I = 400$ Mbits, but its gaps between the "Proposed Solution" are even larger than those of the "Direct Trajectory" scheme when task sizes are large, e.g., $I = 500$ Mbits. All these results verify that the proposed

optimization on bandwidth allocation and UAV's trajectory, as well as making full use of the computing resources at UEs have great effects on minimizing the WSEC of the UAV and UEs. Note that the gaps between the proposed solution and the baselines become larger when I increases, which further indicates that the proposed algorithm is more capable of handling the computation-intensive tasks.

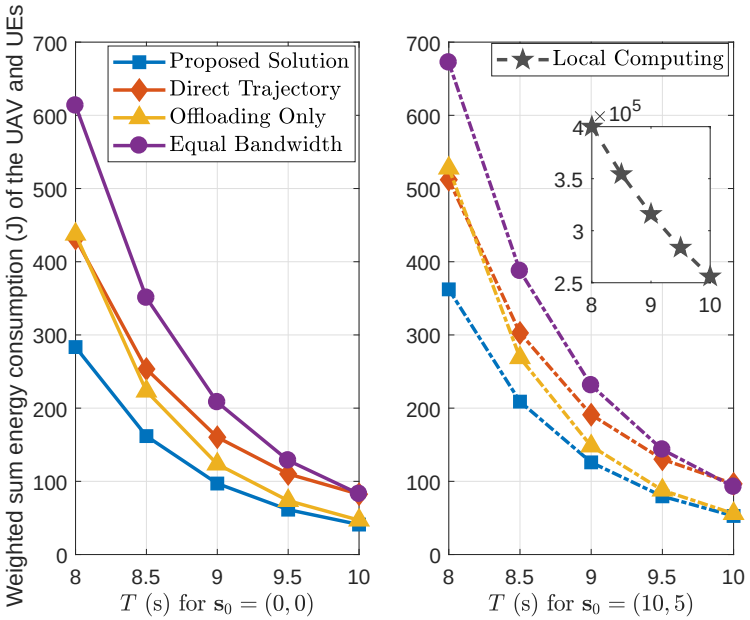


Figure 4.4: The WSEC of the UAV and UEs versus the total task completion time: T (s).

In Figure 4.4, the WSEC w.r.t. the total task completion time T is depicted. We can see that the WSECs of all the schemes decrease with T, coinciding with the intuition that a tradeoff exists between the energy consumption and time consumption for completing the same tasks, and the energy consumption will decrease when the consumed time increases. It is notable that the proposed solution is superior to the four baseline schemes in both scenarios, and the performance improvement is even more prominent with strict time restriction (small T), which further confirms that the proposed algorithm is good at dealing with the latency-critical computation tasks and can achieve a better energy-delay tradeoff. Besides,

some similar insights can also be obtained as from Figure 4.3.

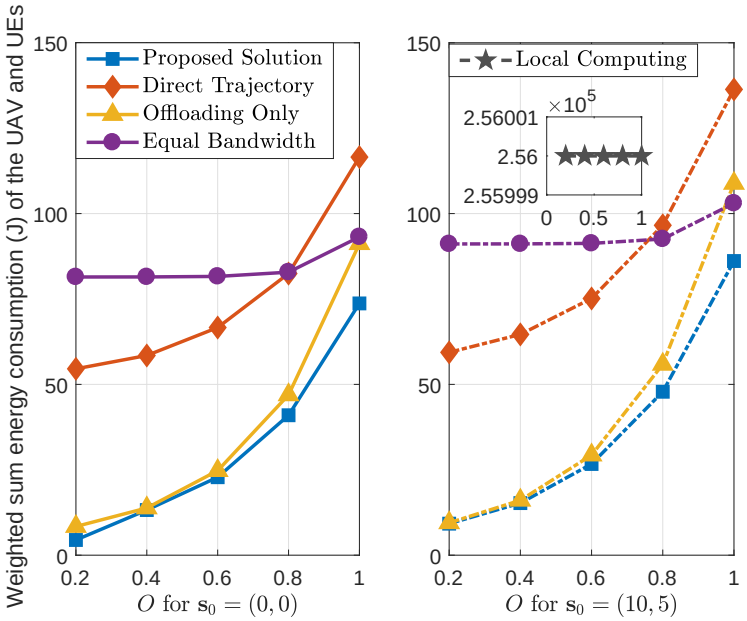


Figure 4.5: The WSEC of the UAV and UEs versus the uniform size ratio of task-output data to task-input data: $O = O_k$ for $k \in \mathcal{K}$.

Figure 4.5 depicts the WSEC w.r.t. the uniform size ratio of the task-output data to the task-input data $O = O_k$ for $k \in \mathcal{K}$. We can see that the proposed scheme outperforms the baselines in both scenarios as in Figure 4.3 and Figure 4.4. The WSEC of the "Local Computing" scheme is constant w.r.t O, while the WSECs of all the other schemes increase with O since more output data will be downloaded to the UEs in the cases with larger O. However, the curves of the "Equal Bandwidth" scheme are almost unchanged for $O \in [0.2, 0.8]$ due to the fact that equally allocated bandwidth to the downloading transmissions should be sufficient to complete the downloading missions, and its performance is much worse than the other offloading schemes for smaller O because of the irrational bandwidth allocation. Note that the gaps between the "Proposed Solution" and the "Direct Trajectory" scheme decrease as O increases since it becomes more difficult to balance the tradeoff between UEs'

task sizes and the relative location of the AP. In comparison, the gaps between the "Proposed Solution" and the "Offloading Only" scheme become large as O increases for the reason that local computing may be an energy-saving way when with a large O. In the scenario of $\mathbf{s}_0 = (10, 5)$, the "Offloading Only" scheme performs even worse than the "Equal Bandwidth" scheme when $O = 1$, which further verifies that the effect of partial local computing in minimizing the WSEC.

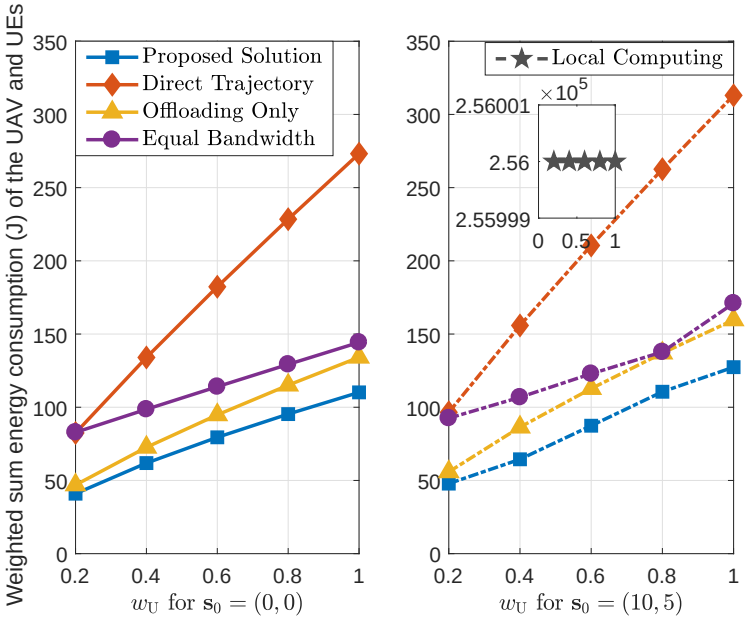


Figure 4.6: The WSEC of the UAV and UEs versus the weight for UAV's energy consumption: w_{U}.

Results for the WSEC versus the UAV's energy consumption weight w_{U} are shown in Figure 4.6. It is clear that the proposed scheme still performs best in both scenarios. All the curves increase with w_{U} except that for "Local Computing" scheme, since a larger proportion of UAV's energy consumption will be calculated into the WSEC with a larger w_{U}. Note that the gaps between the "Proposed Solution" and the "Direct Trajectory" scheme become obviously larger as w_{U} increases in both scenarios especially compared with those gaps related to the

"Offloading Only" and the "Equal Bandwidth" schemes. This is due to the fact that the energy consumption for UAV's propulsion contributes a larger part for WSEC of the "Direct Trajectory" scheme without trajectory optimization, and thus its WSEC increases much faster w.r.t. w_{U} than the other schemes.

From the above results, we can observe that the WSEC for the scenario of $\mathbf{s}_0 = (10, 5)$ is higher than that for the scenario of $\mathbf{s}_0 = (0, 0)$ for all the schemes. It is easy to understand that more energy will be used for UAV's offloading transmission and flying because of the farther average distances between the AP and UEs. The performance of the proposed scheme is also more stable than that of the baseline schemes considering the changing of the relative location of the AP to UEs since its relative WSEC increment is the smallest among the schemes.

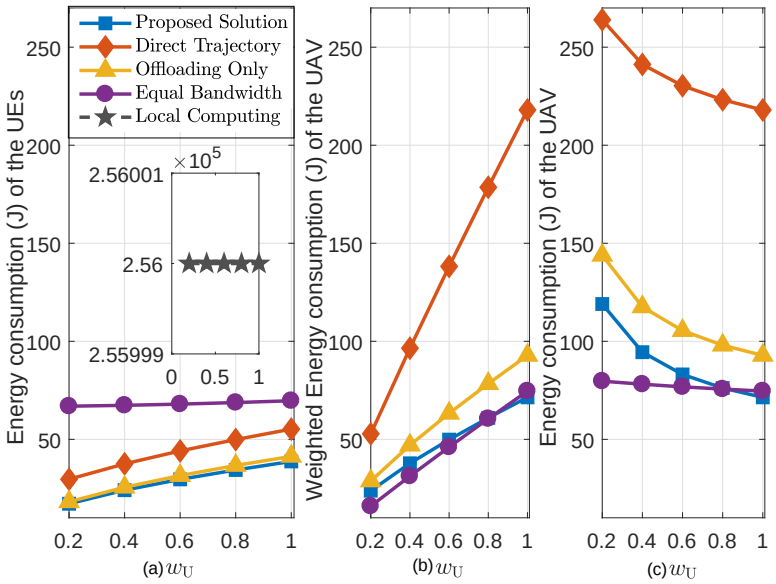


Figure 4.7: Separate energy consumption of the UEs and the UAV versus the weight for UAV's energy consumption: w_{U}.

Based on Figure 4.6, we depict the energy consumption of the UEs (also the weighted energy consumption of the UEs with $w_1 = w_2 = w_3 = w_4 = 1$), the

weighted energy consumption, and the energy consumption of the UAV versus w_{U} in Figure 4.7 (a), (b) and (c), respectively. It is clear that the weighted energy consumption of the UEs and the UAV for the four offloading schemes increase with w_{U} as in (a) and (b), while their energy consumption of the UAV decreases with w_{U} as in (c). This is due to the fact that we aim at minimizing the WSEC, and the objectives increase with w_{U} similar to the results in Figure 4.6. Meanwhile minimizing the UAV's energy consumption becomes more important as w_{U} increases. From this figure, we can better see the tremendous benefits obtained by the UEs from the assistance of the UAV, especially when w_{U} is smaller. In the case of $w_{\mathrm{U}} = 0.2$, the UAV consumes 120 Joule of energy to help the UEs decrease their energy consumption from $2.56 * 10^5$ Joule of the "Local Computing" scheme to 20 Joule of the "Proposed Solution", by providing assistance of task computing and relaying (further offloading to the AP for computing) through the proposed algorithm.

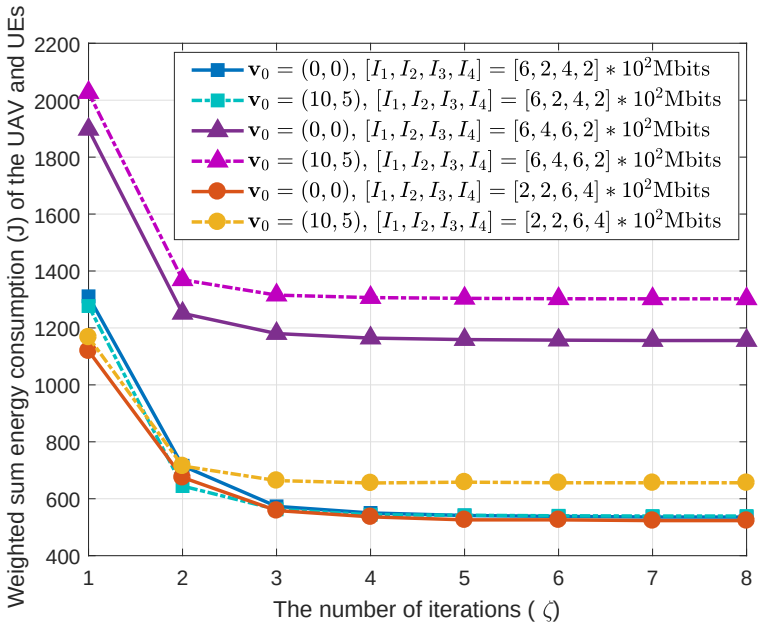


Figure 4.8: The WSEC of the UAV and UEs versus the number of iteration: ζ.

Figure 4.8 shows the WSEC of the proposed solution w.r.t to the iteration index ζ under different settings. From the figure, we can see that the proposed solution almost converges at $\zeta = 3$, i.e., after twice iteration of optimizing $\mathbf{z}$, $\mathbf{B}$ and $\mathbf{u}$, regardless of the UEs' task sizes or the location of the AP.

4.5 Summary

In this chapter, we investigated the UAV-assisted MEC architecture, where the UAV acts as an MEC server and a relay to assist the UEs to compute their tasks or further offload their tasks to the AP for computing. We minimized the WSEC of the UAV and the UEs under some practical constraints, using an alternating algorithm iteratively optimizing the computation resource scheduling, bandwidth allocation, and the UAV's trajectory. The numerical results have confirmed that the UAV's trajectory is greatly affected by the relative location of the AP and the distribution of UEs' task sizes. Besides, significant performance improvement and more stable performance can be achieved by the proposed algorithm over the baseline schemes.

Chapter 5

Mobile Edge Computing in Heterogeneous Cellular Networks with Central Cloud Computing

This chapter is based on our works published in [J3] and [C3] ([72] and [76]).

5.1 Introduction

Most of the existing computing works focused on either the edge or central cloud computing independently, and the edge computing works mainly concentrated on small-scale networks such as the single MEC server or cloudlet case [30, 31, 37, 38, 52–54, 70, 129]. Even though edge computing has been regarded as a promising trend to deal with the ever-growing mobile computing data, it cannot entirely replace the present central cloud computing, due to the fact that edge computing is set to push limited processing and storage capabilities close to UEs but may

be incapable of dealing with big data processing. The latest white paper from ETSI has further illustrated that central cloud computing and edge computing are highly complementary and significant benefits can be attained when utilizing them both [69]. However, the architecture with the coexistence of edge and central cloud has not been thoroughly studied, especially from the communication perspective [14].

Therefore, in this chapter, we study the coexistence and synergy between the edge and central cloud computing in a heterogeneous cellular network (HetNet), which contains a multi-antenna MBS, multiple multi-antenna SBSs and multiple single-antenna UEs. The SBSs are empowered by edge clouds offering limited computing services for UEs, whereas the MBS provides high-performance central cloud computing services to UEs via restricted MIMO backhauls to their associated SBSs. We aim to minimize the system energy consumption used for task offloading and computation by jointly optimizing the cloud selection, the UEs' transmit powers, the SBSs' receive beamformers, and the SBSs' transmit covariance matrices, which is a mixed-integer and non-convex optimization problem. Based on methods such as the decomposition approach and successive pseudoconvex approximation approach, a tractable solution is proposed via an iterative algorithm. The numerical results show that our proposed solution can achieve better performance than conventional schemes using edge or central cloud alone. Also, with large-scale antennas at the MBS, the unique features of massive MIMO backhauls can significantly reduce the complexity of the proposed algorithm and obtain even better performance.

5.2 System Model and Problem Formulation

As shown in Figure 5.1, we consider a two-tier HetNet, where an M-antenna MBS provides wireless MIMO backhauls and is fiber-optic connected to the central cloud with super computing capability, and K SBSs with edge clouds can provide limited computing capabilities.[1] In each small cell, an SBS equipped with L antennas serves a single-antenna UE[2]. Note that existing user association schemes [131] can be adopted to determine which user is connected to an SBS.

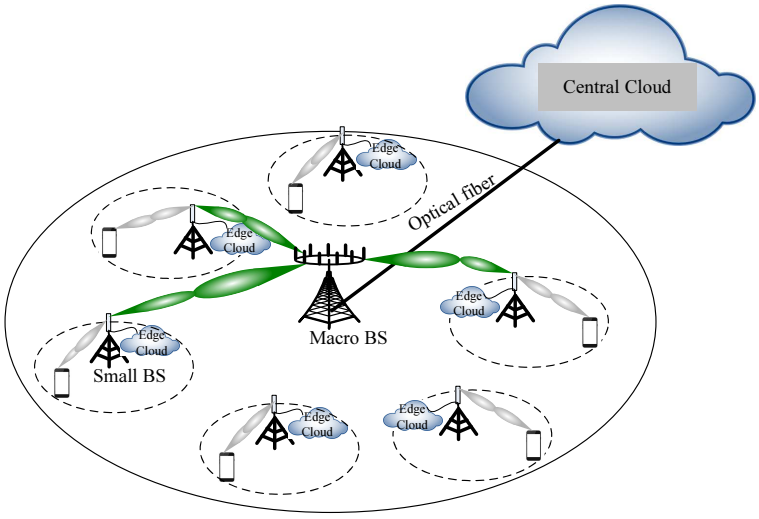


Figure 5.1: An illustration of two-tier HetNets equipped with edge clouds associated with the SBSs and central cloud connected by the MBS via optical fiber, where the MBS provides central cloud computing services for UEs through restricted MIMO/massive MIMO backhauls to their associated SBSs for addressing more complicated computing tasks which cannot be efficiently handled by the SBSs' edge clouds due to the limited computing capabilities.

Let $\mathcal{K} = \{1, \ldots, K\}$ denote the set of the SBSs as well as the UEs. Each UE

[1]The central cloud can be regarded as the computing part of the cloud radio access network (Cloud RAN) [130]. Each edge cloud can be an independent edge computing server co-located at the corresponding SBS or a certain part of computing capability allocated to the SBS from a nearby fiber-optic connected edge computing center [14].

[2]The extended case of serving multiple UEs in each small cell can be effectively dealt with by using the existing orthogonal multiple access techniques for radio resource allocation. In addition, the extended case of our work can be viewed as leveraging equal computing resource sharing at a SBS for multiple active UEs in a small cell, or dedicated computing resource policy for different types of computing services, i.e., each service will be granted one dedicated computing resource.

$k \in \mathcal{K}$ has an atomic highly integrated computation-intensive task, which cannot be partitioned for parallel execution, characterized by a positive tuple $[I_k, O_k, C_k]$. Here I_k is the size (in bits) of the computation task-input data (e.g., the program codes and the input parameters) which cannot be divided and has to be offloaded as a whole either offloaded to and computed at the edge cloud with edge computing mode or offloaded to and computed at the central cloud with central computing mode. $O_k \in (0, 1)$ is the ratio of task-output data size to that of the task-input data, i.e., the output data size should be $O_k I_k$ for UE k, and C_k is the amount of required computing resources for computing 1-bit of UE k's task-input data (i.e., the number of CPU cycles required). The parameters in the task tuple of $[I_k, O_k, C_k]$ can be obtained through task profilers by applying the methods (e.g., call graph analysis) in [7, 14, 105–107].[3] Let B^{a} and B^{b} denote the bandwidths allocated to UEs' access links to their serving SBSs and SBSs' backhaul links to the MBS, respectively. A coordination and monitoring protocol between SBSs and MBS, like the one used in [132, 133], is needed.

Assuming that the UEs are endowed with very limited computing resources, they tend to choose computation offloading to complete their computation tasks remotely, so as to save their own energy and resources. Since the computation tasks offloaded by the UEs could be executed either at the edge clouds or central cloud, the cloud selection needs to be appropriately determined before evaluating the computation latency and energy consumption. Let the binary indicator c_k denote the computing decision, where $c_k = 1$ indicates edge computing, and $c_k = 0$ indicates central cloud computing being selected for each UE $k \in \mathcal{K}$. In the sequel, we

[3]It is assumed that the size of computing outputs, i.e., $O_k I_k$ (a few command bits in our considered scenario in this chapter) is much smaller than I_k (usually measured by Kbit or Mbit) in practice, and thus the downlink overhead such as time and energy consumption for delivering the output data back to the UEs is negligible and can be ignored.

will study the latency and energy consumption of the network, and then formulate the optimization problem for minimizing the network's total energy consumption for task offloading and computation under the central and edge processing latency constraints.

5.2.1 Transmission and Computing Latency

5.2.1.1 Access Transmission Latency

The uplink transmission rate of UE k for offloading the I_k-bit computation tasks to its serving SBS k is expressed as

$$R_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k) = B^{\mathrm{a}} \log_2 \left(1 + \gamma_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k)\right),\ k \in \mathcal{K}, \tag{5.1}$$

with the signal-to-interference-plus-noise ratio (SINR)

$$\gamma_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k) = \frac{p_k^{\mathrm{u}} |\mathbf{w}_k^H \mathbf{h}_{k,k}^{\mathrm{a}}|^2}{\sum_{i=1, i\neq k}^{K} p_i^{\mathrm{u}} |\mathbf{w}_k^H \mathbf{h}_{i,k}^{\mathrm{a}}|^2 + |\mathbf{w}_k^H \mathbf{n}_k|^2}, \tag{5.2}$$

where $\mathbf{w}_k$ is the receive beamforming vector of the k-th SBS, $\mathbf{h}_{i,k}^{\mathrm{a}} \in \mathbb{C}^{L\times 1}$ is the access channel vector between UE i and SBS k, $\mathbf{n}_k$ is a vector of the additive white Gaussian noise with zero mean and variance σ_k^2, and $\mathbf{p}^{\mathrm{u}} \triangleq [p_1^{\mathrm{u}}, \ldots, p_K^{\mathrm{u}}]^T \in \mathbb{R}^{K\times 1}$ denotes the transmit power vector of the UEs. Therefore, the uplink access transmission latency for offloading UE k's task can be calculated as

$$T_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k) = \frac{I_k}{R_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k)},\ k \in \mathcal{K}. \tag{5.3}$$

5.2.1.2 Edge Computing Latency ($c_k = 1$)

Let f_k denote the CPU clock frequency of the k-th edge cloud server associated with SBS k, and thus the corresponding edge computation latency for dealing with the I_k-bits task-input data can be described as

$$T_k^{\text{edge}} = \frac{I_k C_k}{f_k},\ k \in \mathcal{K}, \tag{5.4}$$

which indicates that the value of edge computing latency depends on the offloaded task size (I_k), the required unit computing resource (C_k), and edge cloud's CPU clock frequency (f_k).

5.2.1.3 Central Cloud Processing/Backhaul Transmission Latency ($c_k = 0$)

The central cloud processing latency mainly comes from backhaul transmission and task execution at the central cloud. Due to the central cloud's super computing capability, its computing time is much lower than edge computing, thus we assume that the time for central cloud computing is negligible. Hence, the central cloud processing latency, i.e., the backhaul transmission latency, for the k-th UE can be calculated as[4]

$$T_k^{\text{central}}(\mathbf{Q}) = \frac{I_k}{R_k^{\text{b}}(\mathbf{Q})},\ k \in \mathcal{K}, \tag{5.5}$$

[4]In our considered scenario, the accessing latency of MBS to the central cloud through optical fiber should be negligible especially compared with the wireless backhaul transmission latency. For the extreme case that the optical fiber transmission latency is not negligible, the central cloud processing latency can be re-expressed as $T_k^{\text{central}}(\mathbf{Q}) = \frac{I_k}{R_k^{\text{b}}(\mathbf{Q})} + T_{\text{of}}^{\text{central}}$, where $T_{\text{of}}^{\text{central}}$ is a maximum threshold of optical fiber transmission latency. Even though, the proposed algorithms are still effective.

where $R_k^{\mathrm{b}}(\mathbf{Q})$ is the corresponding backhaul transmission rate given by

$$R_k^{\mathrm{b}}(\mathbf{Q}) = B^{\mathrm{b}} \log_2 \det \left(\mathbf{I} + \Psi(\mathbf{Q}_{-k})^{-1} \mathbf{H}_k^{\mathrm{b}} \mathbf{Q}_k \left(\mathbf{H}_k^{\mathrm{b}} \right)^H \right), \tag{5.6}$$

with the noise-plus-interference covariance matrix

$$\Psi(\mathbf{Q}_{-k}) = \sigma^2 \mathbf{I} + \sum_{i=1, i \neq k}^{N} \mathbf{H}_i^{\mathrm{b}} \mathbf{Q}_i \left(\mathbf{H}_i^{\mathrm{b}} \right)^H. \tag{5.7}$$

In (5.6), $\mathbf{Q}_k$ is the transmit covariance matrix of SBS k, $\mathbf{Q} = \{\mathbf{Q}_k\}_{k=1}^K$ and $\mathbf{Q}_{-k} = \{\mathbf{Q}_i\}_{i=1, i \neq k}^K$ are respectively the compact transmit covariance matrices and the compact transmit covariance matrices except $\mathbf{Q}_k$, and $\mathbf{H}_k^{\mathrm{b}} \in \mathbb{C}^{M \times L}$ is the backhaul channel matrix from SBS k to the MBS. Note that if the computation task of UE $k \in \mathcal{K}$ is executed by the edge cloud of SBS k, i.e. c_k = 1, the transmit covariance matrix at SBS k shall be $\mathbf{Q}_k = \mathbf{0}$.

5.2.2 Energy Consumption

The network energy consumption mainly results from task offloading and task execution/computation. Based on Section 5.2.1, the amount of energy consumption of UE $k \in \mathcal{K}$ for offloading its computation task to its serving SBS k can be calculated as

$$E_k^{\mathrm{a}} = p_k^{\mathrm{u}} T_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k), \ k \in \mathcal{K}. \tag{5.8}$$

If UE k's task is executed by the edge cloud associated with the SBS k, the computation energy consumption at the corresponding edge server is given by

$$E_k^{\text{edge}} = \kappa_k I_k C_k f_k^2,\ k \in \mathcal{K}, \tag{5.9}$$

where κ_k is the effective switched capacitance of the edge cloud k. Else, if the task is executed by the central cloud, we then have the central processing energy consumption, including the backhaul transmission and the computation energy consumption, which is expressed as

$$E_k^{\text{central}} = \text{tr}\left(\mathbf{Q}_k\right) T_k^{\text{central}}(\mathbf{Q}) + \zeta_k E_k^{\text{edge}},\ k \in \mathcal{K}, \tag{5.10}$$

where ζ_k is the ratio of the central cloud's computation energy consumption to that of the edge cloud k for computing the same UE k's task.[5] Thus, the network's total energy consumption for task offloading and computation can be calculated as[6]

$$E_{\text{total}} = \sum_{k=1}^{K} \left(E_k^{\text{a}} + c_k E_k^{\text{edge}} + \left(1 - c_k\right) E_k^{\text{central}} \right). \tag{5.11}$$

5.2.3 Problem Formulation

Our aim is to minimize the network's total energy consumption used for task offloading and computation under central/backhaul and edge processing latency constraints through jointly optimizing UEs' cloud selection decisions in $\mathbf{c} =$

[5] ζ_k can be determined by κ_k, f_k, and the effective switched capacitance and the CPU frequency of the central cloud used for computing UE k's task. Different values of $\{\zeta_k, k \in \mathcal{K}\}$ represent different relationships between the computing energy consumption at central cloud and edge clouds, and may have different effects on edge/central cloud selection and system performance.

[6] Here, the static energy consumption of UEs, SBSs, and MBS consumed by the circuit or cooling is ignored since it has negligible effects on our design.

$\{c_k\}_{k=1}^K$, UEs' transmit power vector $\mathbf{p}^{\mathrm{u}}$, SBSs' receive beamformers in $\mathbf{w} = \{\mathbf{w}_k\}_{k=1}^K$, and SBSs' transmit covariance matrices in $\mathbf{Q}$. To this end, the problem is formulated as

$$\begin{aligned} \min_{\mathbf{c},\mathbf{p}^{\mathrm{u}},\mathbf{w},\mathbf{Q}} \quad & E_{\mathrm{total}} \\ \text{s.t.} \quad & \mathrm{C1}: c_k \in \{0,1\}, \ \ \forall k \in \mathcal{K}, \\ & \mathrm{C2}: (1-c_k)\, T_k^{\mathrm{central}}(\mathbf{Q}) \leq \alpha T_k^{\mathrm{edge}}, \ \ \forall k \in \mathcal{K}, \\ & \mathrm{C3}: T_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k) + c_k T_k^{\mathrm{edge}} \leq T_{\mathrm{th}}, \ \ \forall k \in \mathcal{K}, \\ & \mathrm{C4}: 0 \leq p_k^{\mathrm{u}} \leq P_{\max}^{\mathrm{u}}, \ \ \forall k \in \mathcal{K}, \\ & \mathrm{C5}: \mathbf{Q}_k \succeq \mathbf{0}, \ \ \forall k \in \mathcal{K}. \end{aligned} \tag{5.12}$$

In problem (5.12), $\mathrm{C2}$ represents the central/backhaul processing latency constraint, indicating that the central cloud is selected, i.e., the backhaul is allowed to be used for task offloading, only when the set parameters can make sure that the central/backhaul processing latency is lower than a certain percentage, e.g., α, of edge computing latency. Considering the scarce backhaul resources, this constraint is reasonable in practice and of great benefit to guarantee the high-speed backhaul transmission, avoid the abuse of backhauls, and alleviate the backhaul congestion. Here, $0 < \alpha < 1$ is a predefined ratio parameter for a specified scenario depending on the central cloud and backhaul restriction. For the special case of $\alpha = 0$, the central cloud becomes unavailable as indicated in $\mathrm{C2}$ and thus $c_k = 1$ for $k \in \mathcal{K}$, then problem (5.12) reduces to resource allocation problem in traditional MEC networks, which has been studied from different perspectives in the literature such as [30, 31, 37, 38, 52–55, 70, 129]. $\mathrm{C3}$ is the latency constraint for edge processing, such that the sum of the access transmission latency and the edge computing latency

should not exceed a given threshold $T_{\rm th}$. Note that $T_k^{\rm edge}$ expressed in (5.4) increases with the task size I_k, and thus if edge cloud cannot meet its latency constraint in $\rm C3$ when encounters large tasks, e.g., $T_k^{\rm edge} > T_{\rm th}$, central cloud will be the only option to be utilized, which further indicates the complementary relationship between edge and central cloud computing [69]. $\rm C4$ and $\rm C5$ guarantee the non-negativeness of the transmit power values.

In our considered scenario, we assume that UEs' tasks have already been synchronized. In fact, our work can be easily extended into the cases considering the latency of synchronizing UEs' tasks. For the case with deterministic task arrival model [14], the edge processing latency constraints $\rm C3$ should be changed into $T_k^{\rm syn} + T_k^{\rm a}(\mathbf{p}^{\rm u}, \mathbf{w}_k) + c_k T_k^{\rm edge} \leq T_{\rm th}, k \in \mathcal{K}$, where $T_k^{\rm syn}$ is the synchronization latency of UE k. For the case with random task arrival model [14], we can introduce a maximum synchronization latency threshold, denoted as $T_{\rm syn}$. Then constraints $\rm C3$ can be changed into $T_k^{\rm a}(\mathbf{p}^{\rm u}, \mathbf{w}_k) + c_k T_k^{\rm edge} \leq T_{\rm th} - T_{\rm syn}, k \in \mathcal{K}$. In this way, we can also leverage the algorithms proposed in section 5.3 to solve the corresponding formulated problem (5.12) for minimizing the network's total energy consumption.

5.3 Algorithm Design

The considered problem (5.12) is a mixed-integer and non-convex optimization problem because of the integer cloud selection indicator $\mathbf{c}$, and the non-convex objective function and constraints $\rm C2$, $\rm C3$, which is NP-hard in general and its optimal solution is difficult to achieve. To be tractable, we first need to determine whether edge or central cloud computing will be employed for each UE, and then we can optimize the transmit powers, receive beamformers, and covariance matrices. Hence, a tractable decomposition approach can be developed to solve (5.12) in

an iterative manner considering the fact that $\mathbf{c}$ and $\{\mathbf{p}^{\mathrm{u}}, \mathbf{w}, \mathbf{Q}\}$ are coupled in the objective function and constraints $\mathrm{C}2$, $\mathrm{C}3$ of problem (5.12).

5.3.1 Edge or Central Cloud Computing

As mentioned in section 5.2.3, when the k-th edge cloud's computing time T_k^{edge} is greater than the maximum allowable time T_{th}, the use of edge cloud is infeasible and central cloud computing has to be utilized for UE k, i.e., $c_k = 0$. Next, we optimize the cloud selection indicator $\mathbf{c}$ for the case of $T_k^{\mathrm{edge}} < T_{\mathrm{th}}$ for $k \in \mathcal{K}$. To properly deal with the integer optimization caused by c_k, we first relax $c_k \in \{0, 1\}$ as $\widehat{c}_k \in [0, 1]$, and denote $\widehat{\mathbf{c}} = \{\widehat{c}_k\}_{k=1}^K$ as the set of the relaxed cloud selection variable $\widehat{c}_k$. Then problem (5.12) with given feasible $\{\mathbf{p}^{\mathrm{u}}, \mathbf{w}, \mathbf{Q}\}$ can be decomposed into the following relaxed version

$$\begin{aligned} \min_{\widehat{\mathbf{c}}} \ & \sum_{k=1}^{K} \left(\widehat{c}_k E_k^{\mathrm{edge}} + (1 - \widehat{c}_k)\, E_k^{\mathrm{central}} \right) \\ \text{s.t.} \ & \widehat{\mathrm{C}}1 : \widehat{c}_k \in [0, 1]\,, \ \ \forall k \in \mathcal{K}, \\ & \widehat{\mathrm{C}}2 : (1 - \widehat{c}_k)\, T_k^{\mathrm{central}}(\mathbf{Q}) \leq \alpha T_k^{\mathrm{edge}}, \ \ \forall k \in \mathcal{K}, \\ & \widehat{\mathrm{C}}3 : T_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k) + \widehat{c}_k T_k^{\mathrm{edge}} \leq T_{\mathrm{th}}, \ \ \forall k \in \mathcal{K}. \end{aligned} \tag{5.13}$$

Problem (5.13) is one-dimensional linear programming, and its solution can be given in the following two cases:

- Case 1: Without loss of generality, if the energy consumption of edge computing is lower than that of central processing for UE k's task, i.e, $E_k^{\mathrm{edge}} \leq E_k^{\mathrm{central}}$, the objective function of problem (5.13) is a decreasing function of $\widehat{c}_k$. Therefore, the optimal $\widehat{c}_k^*$ is the maximum value that satisfies

$\widehat{\mathrm{C}}1-\widehat{\mathrm{C}}3$, i.e.,

$$\widehat{c}_k^*=\left[\min\left\{\frac{T_{\mathrm{th}}-T_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}},\mathbf{w}_k)}{T_k^{\mathrm{edge}}},1\right\}\right]^+. \tag{5.14}$$

- Case 2: if $E_k^{\mathrm{edge}} > E_k^{\mathrm{central}}$, the objective function of problem (5.13) is an increasing function of $\widehat{c}_k$, and the optimal $\widehat{c}_k^*$ is the minimum value that satisfies $\widehat{\mathrm{C}}1-\widehat{\mathrm{C}}3$, i.e.,

$$\widehat{c}_k^*=\left[1-\frac{\alpha T_k^{\mathrm{edge}}}{T_k^{\mathrm{central}}(\mathbf{Q})}\right]^+. \tag{5.15}$$

It is seen that the relaxed edge/central cloud computing decision $\widehat{\mathbf{c}}^*$ is reliant on the optimal $\{\mathbf{p}^{\mathrm{u}},\mathbf{w},\mathbf{Q}\}$ of problem (5.12). In the following two subsections, we will focus on obtaining the optimal $\{\mathbf{p}^{\mathrm{u}*},\mathbf{w}^*\}$ and $\mathbf{Q}^*$, respectively, based on a given cloud selection decision $\widehat{\mathbf{c}}$.

5.3.2 UEs' Transmit Powers and SBSs' Receive Beamformers

With a fixed cloud selection decision $\widehat{\mathbf{c}}$, the optimal $\{\mathbf{p}^{\mathrm{u}*},\mathbf{w}^*\}$ can be obtained by solving a subproblem of (5.12) as follows:

$$\begin{aligned}\min_{\mathbf{p}^{\mathrm{u}},\mathbf{w}}\ & \sum_{k=1}^{K}p_k^{\mathrm{u}}T_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}},\mathbf{w}_k)\\ \text{s.t.}\ & \widehat{\mathrm{C}}3,\quad \mathrm{C}4,\end{aligned} \tag{5.16}$$

where $\widehat{\mathrm{C}}3$ and $\mathrm{C}4$ are the corresponding constraints expressed in problem (5.13) and (5.12), respectively. The subproblem (5.16) is non-convex (over $\mathbf{p}^{\mathrm{u}}$) and its objective function is the weighted sum-of-ratios related to $\mathbf{p}^{\mathrm{u}}$, which is challenging

to solve. Here, we first examine the interplay between UEs' transmit power vector $\mathbf{p}^{\mathrm{u}}$ and SBSs' receive beamformers in $\mathbf{w}$.

Lemma 5.1. *For a given feasible* $\mathbf{p}^{\mathrm{u}}$*, the optimal* $\mathbf{w}_k^*$ *of problem* (5.16) *is given by*

$$\mathbf{w}_k^* = \mathrm{eigvec}\left\{\max\left\{\mathrm{eig}\{(\mathbf{\Omega}_{-k})^{-1}\mathbf{\Omega}_k\}\right\}\right\}, \tag{5.17}$$

where $\mathbf{\Omega}_{-k} = \sigma_k^2\mathbf{I}_L + \sum_{i=1,i\neq k}^{K} p_i^{\mathrm{u}}\mathbf{h}_{i,k}^{\mathrm{a}}(\mathbf{h}_{i,k}^{\mathrm{a}})^H$ *and* $\mathbf{\Omega}_k = p_k^{\mathrm{u}}\mathbf{h}_{k,k}^{\mathrm{a}}(\mathbf{h}_{k,k}^{\mathrm{a}})^H$.

Proof. Based on problem (5.16), we can easily find that each SBS's receive beamformer $\mathbf{w}_k$ aims to maximize the SINR, i.e.,

$$\max_{\mathbf{w}_k} \ \gamma_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k). \tag{5.18}$$

Problem (5.18) can be equivalently rewritten as

$$\max_{\mathbf{w}_k} \ \frac{\mathbf{w}_k^H\mathbf{\Omega}_k\mathbf{w}_k}{\mathbf{w}_k^H\mathbf{\Omega}_{-k}\mathbf{w}_k}. \tag{5.19}$$

Note that problem (5.19) is a generalized eigenvector problem and the optimal $\mathbf{w}_k^*$ is the corresponding eigenvector associated with the largest eigenvalue of the matrix $(\mathbf{\Omega}_{-k})^{-1}\mathbf{\Omega}_k$ [134, 135]. Thus, we obtain the result in (5.17). □

With the help of auxiliary variables $\mathbf{t} = \{t_k\}_{k=1}^K$, problem (5.16) over the UEs' transmit power vector $\mathbf{p}^{\mathrm{u}}$ for fixed $\mathbf{w}$ can be equivalently transformed as

$$\begin{aligned} \min_{\mathbf{p}^{\mathrm{u}},\mathbf{t}} \ & \sum_{k=1}^{K} I_k t_k \\ \text{s.t.} \ & \widetilde{\mathrm{C}}1: \frac{p_k^{\mathrm{u}}}{R_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k)} \leq t_k, \ \forall k \in \mathcal{K}, \end{aligned} \tag{5.20}$$

$$\widetilde{\mathrm{C}}2: \gamma_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k) \geq \tau_k, \ \forall k \in \mathcal{K},$$

$$\widetilde{\mathrm{C}}3: 0 \leq p_k^{\mathrm{u}} \leq P_{\max}^{\mathrm{u}}, \ \forall k \in \mathcal{K},$$

where $\tau_k = 2^{\frac{I_k}{B^{\mathrm{a}}\left(T_{\mathrm{th}} - \widehat{c}_k T_k^{\mathrm{edge}}\right)}} - 1$.

Lemma 5.2. *The optimal solution* $(\mathbf{p}^{\mathrm{u}*}, \mathbf{t}^*)$ *of problem* (5.20) *satisfies the Karush-Kuhn-Tucker (KKT) conditions of the following* K $(k \in \mathcal{K})$ *subproblems*

$$\begin{aligned} \min_{p_k^{\mathrm{u}}} \quad & \left(\lambda_k + M_k\right) p_k^{\mathrm{u}} - \lambda_k t_k R_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k) \\ \text{s.t.} \quad & \widetilde{\mathrm{C}}2: \gamma_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}}, \mathbf{w}_k) \geq \tau_k, \\ & \widetilde{\mathrm{C}}3: 0 \leq p_k^{\mathrm{u}} \leq P_{\max}^{\mathrm{u}}, \end{aligned} \tag{5.21}$$

with

$$\begin{aligned} M_k = & \sum_{j=1, j\neq k}^{K} \lambda_j t_j \frac{B_{\mathrm{a}}}{\ln 2} \frac{\left(\gamma_j^{\mathrm{a}}\right)^2 |\mathbf{w}_j^H \mathbf{h}_{k,j}^{\mathrm{a}}|^2}{p_j^{\mathrm{u}} |\mathbf{w}_j^H \mathbf{h}_{j,j}^{\mathrm{a}}|^2 \left(1 + \gamma_j^{\mathrm{a}}\right)} + \\ & \sum_{j=1, j\neq k}^{K} \mu_j \frac{\left(\gamma_j^{\mathrm{a}}\right)^2 |\mathbf{w}_j^H \mathbf{h}_{k,j}^{\mathrm{a}}|^2}{p_j^{\mathrm{u}} |\mathbf{w}_j^H \mathbf{h}_{j,j}^{\mathrm{a}}|^2}, \end{aligned} \tag{5.22}$$

where $\{\lambda_k\}_{k=1}^K$ *and* $\{\mu_k\}_{k=1}^K$ *are Lagrange multipliers associated with the constraints* $\widetilde{\mathrm{C}}1$ *and* $\widetilde{\mathrm{C}}2$ *of problem* (5.20)*, respectively, and* $M_k = -\sum_{j=1, j\neq k}^{K} \lambda_j t_j \frac{\partial R_j^{\mathrm{a}}}{\partial p_k^{\mathrm{u}}} - \sum_{j=1, j\neq k}^{K} \mu_j \frac{\partial \gamma_j^{\mathrm{a}}}{\partial p_k^{\mathrm{u}}}$. *For optimal* $(\mathbf{p}^{\mathrm{u}*}, \mathbf{w}^*)$*,* λ_k *and* t_k *are respectively calculated as*

$$\lambda_k = \frac{I_k}{R_k^{\mathrm{a}}\left(\mathbf{p}^{\mathrm{u}*}, \mathbf{w}_k^*\right)}, \tag{5.23}$$

$$t_k = \frac{p_k^{\mathrm{u}*}}{R_k^{\mathrm{a}}\left(\mathbf{p}^{\mathrm{u}*}, \mathbf{w}_k^*\right)}. \tag{5.24}$$

Proof. See Appendix C.1. □

Through Lemma 5.2, we know that the optimal solution of problem (5.20) can be obtained by solving K parallel subproblems described in (5.21). Given λ_k and t_k, the subproblem (5.21) is convex w.r.t. p_k^{u}. Therefore, we have the following theorem.

Theorem 5.1. *The solution of subproblem* (5.21) *is given by*

$$p_k^{\mathrm{u}*} = \begin{cases} \dfrac{\tau_k}{\Lambda_k}, & \text{if } G_k < \dfrac{\tau_k}{\Lambda_k}, \\ G_k, & \text{if } \dfrac{\tau_k}{\Lambda_k} \leq G_k \leq P_{\max}^{\mathrm{u}}, \\ P_{\max}^{\mathrm{u}}, & \text{if } G_k > P_{\max}^{\mathrm{u}}, \end{cases} \tag{5.25}$$

$$\mu_k^* = \begin{cases} \dfrac{\lambda_k + M_k}{\Lambda_k} - \dfrac{B_{\mathrm{a}}}{\ln 2}\dfrac{\lambda_k t_k}{\tau_k + 1}, & \text{if } G_k < \dfrac{\tau_k}{\Lambda_k}, \\ 0, & \text{otherwise}, \end{cases} \tag{5.26}$$

$$\nu_k^* = \begin{cases} \dfrac{B_{\mathrm{a}}}{\ln 2}\dfrac{\lambda_k t_k}{P_{\max}^{\mathrm{u}} + 1/\Lambda_k} - \lambda_k - M_k, & \text{if } G_k > P_{\max}^{\mathrm{u}}, \\ 0, & \text{otherwise}, \end{cases} \tag{5.27}$$

where we define $\Lambda_k \triangleq \frac{|\mathbf{w}_k^H \mathbf{h}_{k,k}^{\mathrm{a}}|^2}{\sum_{i=1, i\neq k}^{K} p_i^{\mathrm{u}} |\mathbf{w}_k^H \mathbf{h}_{i,k}^{\mathrm{a}}|^2 + |\mathbf{w}_k^H \mathbf{n}_k|^2}$, $G_k \triangleq \frac{B_{\mathrm{a}}}{\ln 2}\frac{\lambda_k t_k}{\lambda_k + M_k} - \frac{1}{\Lambda_k}$, *and* μ_k^* *and* ν_k^* *are respectively the optimal Lagrange multipliers associated with the constraints* $\widetilde{\mathrm{C}}2$ *and* $\widetilde{\mathrm{C}}3$ *of problem* (5.21).

Proof. See Appendix C.2. □

In light of the results in **Lemma** 5.1, **Lemma** 5.2 and **Theorem** 5.1, we provide an iterative approach to effectively solve problem (5.16) for obtaining UEs' transmit powers and SBSs' receive beamformers, which is shown in **Algorithm** 5.1.

Algorithm 5.1 Solution of Problem (5.16)

1: **Initialize** $p_k^{\mathrm{u}} = P_{\max}^{\mathrm{u}}, \forall k$. Set $\mathbf{w}_k$ based on **Lemma** 5.1.

2: **Repeat**

3: a) Given $\mathbf{w}$, Loop:

i): Compute M_k, λ_k and t_k based on **Lemma** 5.2.

ii): Update p_k^{u} and μ_k based on **Theorem** 5.1.

Until convergence.

4: b) Update $\mathbf{w}$ based on **Lemma** 5.1.

5: Until convergence, and obtain the optimal $\{\mathbf{p}^{\mathrm{u}*}, \mathbf{w}^*\}$.

The convergence of **Algorithm** 5.1 can be guaranteed since the objective function of problem (5.16) decreases with the iteration index (in step 3 and step 4 of **Algorithm** 5.1), which is indicated from optimizing $\mathbf{p}^{\mathrm{u}}$ and $\mathbf{w}$ in each iteration as shown in **Lemma** 5.1 and **Lemma** 5.2, respectively.

5.3.3 SBSs' Transmit Covariance Matrices

With a fixed cloud selection decision $\widehat{\mathbf{c}}$, the optimal $\mathbf{Q}^*$ can be obtained by solving the following subproblem:

$$\begin{aligned} \min_{\mathbf{Q}} \ \ & y\left(\mathbf{Q}\right) = \sum_{k=1}^{K} \left(1-\widehat{c}_k\right) \mathrm{tr}\left(\mathbf{Q}_k\right) T_k^{\mathrm{central}}(\mathbf{Q}) \\ \text{s.t.} \ \ & \widehat{\mathrm{C}}2: R_k^{\mathrm{b}}(\mathbf{Q}) \geq \left(1-\widehat{c}_k\right) \frac{I_k}{\alpha T_k^{\mathrm{edge}}}, \ \forall k \in \mathcal{K}, \ \ \mathrm{C}5, \end{aligned} \tag{5.28}$$

where $\widehat{\mathrm{C}}2$ and $\mathrm{C}5$ are the corresponding constraints expressed in problem (5.13) and (5.12), respectively, and $\widehat{\mathrm{C}}2$ is re-expressed in an equivalent form here. Problem (5.28) is non-convex due to the non-convexity of the objective function and constraint $\widehat{\mathrm{C}}2$, which cannot be solved directly. Thus, we resort to a successive

pseudoconvex approach to solve this problem iteratively, which has many advantages such as fast convergence and parallel computation [136].

First, let $\mathbf{Q}^l$ denote the $\mathbf{Q}$ value in the l-th iteration. Thus the non-convex item $(1-\widehat{c}_k)\mathrm{tr}\left(\mathbf{Q}_k\right) T_k^{\mathrm{central}}(\mathbf{Q})$ for each $k \in \mathcal{K}$ in the objective function can be approximated as a pseudoconvex function at $\mathbf{Q}^l$, which is written as

$$\widehat{y}_k(\mathbf{Q}_k;\mathbf{Q}^l) \triangleq (1-\widehat{c}_k)\,\frac{I_k \mathrm{tr}(\mathbf{Q}_k)}{R_k^{\mathrm{b}}(\mathbf{Q}_k;\mathbf{Q}^l)} + \chi_k(\mathbf{Q}_k), \tag{5.29}$$

where $\chi_k(\mathbf{Q}_k) = \sum_{j=1,j\neq k}^{K}(1-\widehat{c}_j) I_j \mathrm{tr}(\mathbf{Q}_j^l)\left\langle (\mathbf{Q}_k - \mathbf{Q}_k^l), \nabla_{\mathbf{Q}_k} \frac{1}{R_j^{\mathrm{b}}(\mathbf{Q}^l)} \right\rangle$ with $\langle \mathbf{A}_1, \mathbf{A}_2 \rangle \triangleq \Re\{\mathrm{tr}(\mathbf{A}_1^H \mathbf{A}_2)\}$ is a function obtained by linearizing the non-convex function $\sum_{j=1,j\neq k}^{K}(1-\widehat{c}_j)\mathrm{tr}\left(\mathbf{Q}_j\right) T_j^{\mathrm{central}}(\mathbf{Q})$ in $\mathbf{Q}_k$ at the point $\mathbf{Q}^l$, and $\nabla_{\mathbf{Q}_k} \frac{1}{R_j^{\mathrm{b}}(\mathbf{Q}^l)}$ is the Jacobian matrix of $\frac{1}{R_j^{\mathrm{b}}(\mathbf{Q}^l)}$ w.r.t. $\mathbf{Q}_k$ at the point $\mathbf{Q}^l$. Based on (5.29), we can approximate the objective function $y\left(\mathbf{Q}\right)$ of problem (5.28) at $\mathbf{Q}^l$ as

$$\widehat{y}(\mathbf{Q};\mathbf{Q}^l) = \sum_{k=1}^{K} \widehat{y}_k(\mathbf{Q}_k;\mathbf{Q}^l). \tag{5.30}$$

It is easily seen that $\widehat{y}(\mathbf{Q};\mathbf{Q}^l)$ is pseudoconvex and has the same gradient with $y\left(\mathbf{Q}\right)$ at $\mathbf{Q}=\mathbf{Q}^l$. Hence, converging to a stationary point is guaranteed for the successive pseudoconvex approach [136].

Then, we equivalently rewrite the non-concave function $R_k^{\mathrm{b}}(\mathbf{Q})$ in constraint $\widehat{\mathrm{C}}2$ as a difference of two concave functions as expressed in (5.31a) according to its definition in (5.6). By leveraging the first-order Taylor expansion at $\mathbf{Q}^l$, the second concave function denoted as $R_k^{\mathrm{b}2}(\mathbf{Q}) = B^{\mathrm{b}} \log_2 \det\left(\sigma^2\mathbf{I} + \sum_{i=1,\neq k}^{K} \mathbf{H}_i^{\mathrm{b}} \mathbf{Q}_i \left(\mathbf{H}_i^{\mathrm{b}}\right)^H\right)$ can be approximated by its linear up-

per bound. Hence, $R_k^{\mathrm{b}}(\mathbf{Q})$ can be approximated as

$$R_k^{\mathrm{b}}(\mathbf{Q}) = B^{\mathrm{b}} \log_2 \det\left(\sigma^2 \mathbf{I} + \mathbf{\Xi}(\mathbf{Q})\right) - R_k^{\mathrm{b}2}(\mathbf{Q}) \tag{5.31a}$$

$$\geqslant B^{\mathrm{b}} \log_2 \det\left(\sigma^2 \mathbf{I} + \mathbf{\Xi}(\mathbf{Q})\right) - R_k^{\mathrm{b}2}(\mathbf{Q}^l) - \sum_{j=1, j\neq k}^{K} \left\langle (\mathbf{Q}_j - \mathbf{Q}_j^l), \nabla_{\mathbf{Q}_j} R_k^{\mathrm{b}2}(\mathbf{Q}^l) \right\rangle \triangleq \bar{R}_k^{\mathrm{b}}(\mathbf{Q}), \tag{5.31b}$$

where $\mathbf{\Xi}(\mathbf{Q}) = \sum_{i=1}^{K} \mathbf{H}_i^{\mathrm{b}} \mathbf{Q}_i \left(\mathbf{H}_i^{\mathrm{b}}\right)^H$. Here, $\bar{R}_k^{\mathrm{b}}(\mathbf{Q})$ expressed in (5.31b) is a concave function over $\mathbf{Q}$.

Therefore, at point $\mathbf{Q}^l$, the original problem (5.28) can be approximately transformed as

$$\begin{aligned} &\min_{\mathbf{Q}} \ \ \widehat{y}(\mathbf{Q}; \mathbf{Q}^l) \\ &\text{s.t. } \ \overline{\mathrm{C}}2 : \bar{R}_k^{\mathrm{b}}(\mathbf{Q}; \mathbf{Q}^l) \geq (1 - \widehat{c}_k) \frac{I_k}{\alpha T_k^{\mathrm{edge}}}, \ \forall k \in \mathcal{K}, \ \ \mathrm{C}5. \end{aligned} \tag{5.32}$$

The objective function of problem (5.32) is a sum of K pseudoconvex functions each containing a fractional function and a linear function. In addition, all the constraints in problem (5.32) are convex. Hence, by leveraging the Dinkelbach-like algorithm [137] and introducing a set of auxiliary variables for the K fractional functions in the objective function, problem (5.32) can be transformed into a solvable convex optimization problem, which can be effectively solved by CVX [128] and owns provable convergence [136]. Let $\mathbb{B}\mathbf{Q}^l$ represent the optimal solution of problem (5.32) at the l-th iteration, and thus the value of $\mathbf{Q}$ in the next $(l+1)$-th iteration can be updated as

$$\mathbf{Q}^{l+1} = \mathbf{Q}^l + \varsigma(l)(\mathbb{B}\mathbf{Q}^l - \mathbf{Q}^l), \tag{5.33}$$

where $\varsigma(l)$ is the step size at the l-th iteration and can be obtained through the successive line search, and $\mathbb{B}\mathbf{Q}^l - \mathbf{Q}^l$ is the descent direction of $y(\mathbf{Q})$ [136]. Thus, the solution of problem (5.28) can be iteratively obtained.

Based on the aforementioned analysis of optimizing the variables $\{\mathbf{p}^{\mathrm{u}*}, \mathbf{w}^*, \mathbf{Q}^*\}$, **Algorithm** 5.2 is proposed to solve the original problem (5.12) for minimizing the network's total energy consumption by jointly optimizing $\mathbf{c}$, $\mathbf{p}^{\mathrm{u}}$, $\mathbf{w}$, and $\mathbf{Q}$.

5.3.4 Convergence and Complexity

The convergence of **Algorithm** 5.2 is easy to prove in light of the guaranteed convergence of **Algorithm** 5.1, the successive pseudoconvex method and the Dinkelbach-like algorithm used to solve problem (5.32) [136, 137], and the update process of the cloud selection $\widehat{\mathbf{c}}$ illustrated in Section 5.3.1. Note that the objective function of problem (5.12), i.e., the network's total energy consumption for task offloading and computation, is a decreasing function of the iteration index (in step 3 and step 4 of **Algorithm** 5.2), which ensures the convergence of **Algorithm** 5.2.

The proposed **Algorithm** 5.2 enjoys an acceptable complexity as well as an easy implementation. In each iteration, the majority of computational complexity lies in solving subproblem (5.20) for obtaining the optimal $\mathbf{p}^{\mathrm{u}*}$ and the approximate subproblem (5.32) for obtaining the optimal $\mathbf{Q}^*$ with a given $\widehat{\mathbf{c}}$. In the proposed algorithm, problem (5.20) can be equivalently transformed into K independent subproblems (5.21) and thus can be easily solved in a parallel way. Moreover, the optimal solution of each subproblem has closed-form expressions as indicated in **Theorem** 5.1, which only generates a complexity ordered by $\mathcal{O}(K)$. For the approximate subproblem (5.32) of obtaining $\mathbf{Q}^*$, the Dinkelbach-like algorithm is

Algorithm 5.2 Solution of Problem (5.12)

1: **Initialize** $p_k^{\mathrm{u}} = P_{\max}^{\mathrm{u}}, \forall k$. Set $\mathbf{w}_k$ based on **Lemma** 5.1.

Based on the constraint $\widehat{\mathrm{C}}3$ of problem (5.13), we first set the initial $\widehat{c}_k = \left[\min\left\{\frac{T_{\mathrm{th}}-T_k^{\mathrm{a}}(\mathbf{p}^{\mathrm{u}},\mathbf{w}_k)}{T_k^{\mathrm{edge}}}, 1-\delta\right\}\right]^+$, where $\delta \in (0, 0.5)$ is a tolerant value to avoid the selection of solely edge clouds or central cloud at the initial point. Then, based on the constraint $\widehat{\mathrm{C}}2$ of problem (5.13), $\mathbf{Q}$ is set to meet $T_k^{\mathrm{central}}(\mathbf{Q}) = \frac{\alpha T_k^{\mathrm{edge}}}{1-\widehat{c}_k}$ through the use of ZF precoding with equal power allocation at each SBS.

2: **Repeat**

3: a) Given $\{\widehat{c}_k\}_{k=1}^K$:

i): Update $\{\mathbf{p}^{\mathrm{u}}, \mathbf{w}\}$ based on **Algorithm** 5.1.

ii): Loop:

ii-1): Solve problem (5.32) via Dinkelbach-like algorithm [137].

ii-2): Update $\mathbf{Q}^l$ based on (5.33).

Until convergence, and obtain the updated $\mathbf{Q}$.

4: b) Update $\{\widehat{c}_k\}_{k=1}^K$ according to subsection 5.3.1.

5: Until convergence, and obtain solution $\{\mathbf{c}^*, \mathbf{p}^{\mathrm{u}*}, \mathbf{w}^*, \mathbf{Q}^*\}$, in which $\mathbf{c}^*$ is obtained by rounding the cloud selection solution of problem (5.13), i.e., $\widehat{\mathbf{c}}$, and $\mathbf{p}^{\mathrm{u}*}, \mathbf{w}^*, \mathbf{Q}^*$ are obtained based on the final obtained $\mathbf{c}^*$.

proved to exhibit a linear convergence rate [137] and the corresponding convex optimization problem can be efficiently solved by the software CVX [128], thus the generated complexity is acceptable in general.

The offloading/transmissions in the previously mentioned scenario with traditional MIMO backhauls can be implemented by leveraging the Sub-6 GHz frequency band. Note that the real-time implementation of proposed **Algorithm** 5.2 is achievable if the number of SBSs, UEs is not very large. However, due to the iterative property of **Algorithm** 5.2, the real-time implementation may be hindered by the increasing computational complexity as the number of SBSs, UEs increases. One promising way to overcome this drawback is to leverage the deep learning method. Specifically, the proposed **Algorithm** 5.2 can be utilized to generate the required data samples and train the deep neural networks (DNNs) offline, and then the well-trained DNNs is capable of emulating **Algorithm** 5.2 and inferencing the obtained solution online to realize real-time implementation.

In order to further reduce the computational complexity of solving the optimization problem for minimizing the network's total energy consumption of task offloading and computation, we will consider the scenario with massive MIMO backhauls in the following section by applying the massive MIMO technology at the MBS. It demonstrates that the complexity of the proposed algorithm can be substantially reduced while even better performance can be achieved compared to the case with traditional MIMO backhauls.

5.4 Massive MIMO Backhauls

In the prior sections, we have studied the synergy of combining edge-central cloud computing with traditional multi-cell MIMO backhauls. Since massive MIMO

has been one of the key 5G radio-access technologies, in this section, we further consider the time-division duplex (TDD) massive MIMO aided backhauls in the Rayleigh fading environment, i.e., the MBS is equipped with a very large number of antennas and the SBSs only use one single transmit antenna ($M \gg K$).

There are two main merits for massive MIMO backhaul transmission:

1) Since SBSs and MBSs are usually still and the backhaul channels will become deterministic, a phenomenon known as "channel hardening" [138, 139], and thus the backhaul channel coherence time will be much longer than ever before, which means that the time spent on uplink channel estimation will be much lower. Some real-time massive MIMO channel measurement works such as [140] also demonstrated that the use of massive antennas can mitigate the fast-fade error bursts, and enable much less frequent update of power control in low-mobility environments compared to the single-antenna case (see [140, Fig. 8]);

2) As shown in [141], simple linear processing methods can achieve nearly-optimal performance. As a result, we will consider two linear detection schemes at the MBS with massive antennas, namely the maximal-ratio combining (MRC) and the zero-forcing (ZF), to provide low-complexity massive MIMO backhaul solutions.

5.4.1 MRC Receiver at the MBS

When MRC receiver is applied at the MBS, we consider a lower-bound achievable backhaul rate for tractability, which can well approximate the exact massive MIMO transmission rate as confirmed in [142]. As such, given the cloud selection decision

$\widehat{\mathbf{c}}$, the backhaul related problem (5.28) reduces to

$$\begin{aligned} \min_{\mathbf{q}} \quad & \sum_{k=1}^{K} (1-\widehat{c}_k)\, q_k \frac{I_k}{R_k^{\mathrm{b}}(\mathbf{q})} \\ \text{s.t.} \quad & \widehat{\mathrm{C}}2: R_k^{\mathrm{b}}(\mathbf{q}) \geq (1-\widehat{c}_k)\, \frac{I_k}{\alpha T_k^{\mathrm{edge}}},\ \forall k \in \mathcal{K}, \\ & \mathrm{C}5: q_k \geq 0,\ \forall k \in \mathcal{K}, \end{aligned} \tag{5.34}$$

where q_k is the k-th SBS's transmit power, $\mathbf{q} = [q_1, \cdots, q_K]$, and

$$R_k^{\mathrm{b}}(\mathbf{q}) = B^{\mathrm{b}} \log_2 \Big(1 + (M-1)\, \frac{q_k \beta_k}{\sum_{i=1, i\neq k}^{K} q_i \beta_i + \sigma_k^2}\Big), \tag{5.35}$$

in which β_i is the large-scale fading coefficient of the link between SBS i and the MBS [142]. Problem (5.34) is non-convex, but can be equivalent to problem (5.16) with $\mathbf{w}_k = 1$. Thus, it can be directly solved by using **Algorithm** 5.1. Note that when using **Algorithm** 5.1, SBSs' initial feasible transmit power vector $\mathbf{q}$ needs to be carefully selected. Here, we assume that the present fractional power control solution applied in 3GPP-LTE [143] can satisfy the constraint $\widehat{\mathrm{C}}2$ in (5.34), i.e., $q_k = (d_k)^{\epsilon \varpi^{\mathrm{b}}}$, where d_k is the communication distance between the k-th SBS and the MBS, $\epsilon \in [0, 1]$ is the pathloss compensation factor, and ϖ^{b} is the pathloss exponent of the backhaul links. For the special case of full compensation ($\epsilon = 1$), the number of MBS's antennas needs to meet

$$M \geq 1 + (K-1) \left(2^{\frac{(1-\widehat{c}_k) I_k}{B^{\mathrm{b}} \alpha T_k^{\mathrm{edge}}}} - 1 \right). \tag{5.36}$$

5.4.2 ZF Receiver at the MBS

When ZF receiver is applied at the MBS, we adopt the corresponding tight lower-bound achievable rate shown in [142]. Given the cloud selection decision $\widehat{\mathbf{c}}$, the backhaul related problem (5.28) reduces to the following version

$$\begin{aligned} \min_{\mathbf{q}} \ & \sum_{k=1}^{K} (1-\widehat{c}_k) \frac{q_k I_k}{R_k^{\mathrm{b}}(q_k)} \\ \text{s.t.} \ & \widehat{\mathrm{C}}2: R_k^{\mathrm{b}}(q_k) \geq (1-\widehat{c}_k) \frac{I_k}{\alpha T_k^{\mathrm{edge}}}, \ \forall k \in \mathcal{K}, \\ & \mathrm{C}5: q_k \geq 0, \ \forall k \in \mathcal{K}, \end{aligned} \tag{5.37}$$

where $R_k^{\mathrm{b}}(q_k) = B^{\mathrm{b}} \log_2 \left(1 + (M-K) \frac{q_k \beta_k}{\sigma_k^2}\right)$. Since $\frac{q_k}{R_k^{\mathrm{b}}(q_k)}$ is an increasing function of q_k according to the derivative $\frac{\partial\left(\frac{q_k}{R_k^{\mathrm{b}}(q_k)}\right)}{\partial q_k} \geq 0$, the optimal q_k^* is the minimum value that meets the constraints $\widehat{\mathrm{C}}2$ and $\mathrm{C}5$ in (5.37), i.e.,

$$q_k^* = \frac{2^{\frac{(1-\widehat{c}_k) I_k}{B^{\mathrm{b}} \alpha T_k^{\mathrm{edge}}}} - 1}{(M-K) \frac{\beta_k}{\sigma_k^2}}, \ \forall k \in \mathcal{K}. \tag{5.38}$$

Based on the above analysis, when massive MIMO backhauls are employed at the MBS, the solution of problem (5.12) can still be obtained by using the proposed **Algorithm** 5.2, where the optimal SBSs' transmit powers are given by the solution of problem (5.34) for the MRC receiver or (5.38) for the ZF receiver.

In comparison with the case of using traditional MIMO backhaul, the MRC and ZF linear detection schemes for the case with massive MIMO backhaul links can enjoy super-low complexity. For MRC scheme, the problem (5.34) can be effectively solved by **Algorithm** 5.1, and its computational complexity is with the order of $\mathcal{O}(K)$. For ZF scheme, the closed-form solution of problem (5.37)

can be directly obtained, and its complexity order is $\mathcal{O}(1)$. Hence, applying the massive MIMO technology at the MBS can significantly facilitate the cooperation between the edge and central clouds by providing easier but more efficient backhaul offloading for UEs to access the central cloud computing services.

In the scenario with the massive MIMO backhauls, the real-time online implementation of the proposed **Algorithm** 5.2 is more achievable in general especially considering the case with ZF receiver at the MBS, where the closed-form solution of the SBS's offloading power to the cental cloud can be obtained and better performance can be achieved as well. In addition, the data-driven approach with offline trained DNNs can be further leveraged to achieve real-time online implementations even in the scenario with massively connected user devices. It should be noted that we currently consider the offloading/transmissions where the massive MIMO backhauls are implemented through the Sub-6 GHz frequency band. Actually, the performance can be further enhanced if we combine the technology of massive MIMO with the technology of mmWave communications, which is regarded as a potential extension of our current work.

5.5 Numerical Results

In this section, simulation results are presented to evaluate the performance of the proposed algorithms and shed light on the effects of the key parameters including the ratio of energy consumption between central and edge cloud computing ($\zeta_k = \zeta, k \in \mathcal{K}$), the task size ($I_k = I, k \in \mathcal{K}$), the latency threshold of edge processing ($T_{\rm th}$), the required ratio parameter of edge computing time for backhaul transmission (α), and the edge clouds' CPU clock frequency ($f_k = f, k \in \mathcal{K}$). The performance of some practical schemes are also given as benchmarks, including

the "Edge-cloud-only", "Central-cloud-only" schemes, and a scheme with fixed cloud selection, denoted as "Half edge, Half central" scheme where half number of UEs choose edge cloud and the other half use central cloud to complete their computation tasks. Besides, the "Initial feasible solution", representing the case with the initial values set in **Algorithm** 5.2, is also given as a baseline to show the performance improvement of optimizing the crucial system parameters. Note that the performance indicators (the total energy consumption and the percentage of UEs that select edge cloud computing) shown in the following figures are averaged over 500 independent channel realizations.

All the small-scale fading channel coefficients follow independent and identically complex Gaussian distribution with zero mean and unit variance. The pathloss between SBSs and UEs and between MBS and SBSs are respectively set as $-(140.7 + 36.7\log_{10} d)$dB and $-(100.7 + 23.5\log_{10} d)$dB according to 3GPP TR 36.814 [144], where d (in kilometer) is the distance between two nodes. In the following simulation results, it is assumed that the MBS is located at the origin of the horizontal coordinate system, where the coverage area of the macro cell is a MBS-centered circle with the radius of r^{b}. The locations of the SBSs are randomly deployed within the MBS-centered circle area with the radius of $r^{\mathrm{b}} - r^{\mathrm{a}}$, and the location of the UE in each small cell is randomly generated within the SBS-centered circle area with the radius of r^{a}. With the location information of the MBS, SBSs and UEs, we can then easily calculate the distance between two specific nodes. The other basic simulation parameters are listed in **Table** 5.1.

Table 5.1: Simulation Parameters

Parameter	**Symbol**	**Value**
Bandwidth for an access or backhaul links	B^{a},B^{b}	10 MHz
Noise power spectral density for an access or backhaul links	$\sigma_k^2, k \in \mathcal{K},\ \sigma^2$	-174 dBm/Hz
Pathloss exponent for access links	ϖ^{a}	3.67
Pathloss exponent for backhaul links	ϖ^{b}	2.35
Pathloss compensation factor	ϵ	1
Radius of the small cells	r^{a}	50 m
Radius of the macro cell	r^{b}	500 m
Number of SBSs/UEs	K	6
Number of antennas for each SBS	L	2
UEs' maximum transmit power	$P_{\max}^{\mathrm{u}}$	23 dBm
Required CPU cycles per bit	$C_k,\ k \in \mathcal{K}$	300 cycles/bit
The effective switched capacitance of the SBSs' processors	$\kappa_k,\ k \in \mathcal{K}$	10^{-28}
The tolerant value in **Algorithm** 5.2	δ	0.1

5.5.1 Improvement with Traditional MIMO Backhauls

In this subsection, numerical results for the integrated edge and central cloud computing system with traditional MIMO backhauls are presented in comparison with the benchmarks mentioned before. These results can properly demonstrate the performance enhancement of using the proposed algorithm through jointly optimizing the key system parameters including cloud selection decision, UEs' transmit powers, SBSs' receive beamformers, and SBSs' transmit covariance matrices.

Figure 5.2 shows the effect of the uniform computing energy ratio $\zeta = \zeta_k, k \in \mathcal{K}$ on the total energy consumption of the system with traditional MIMO backhauls. We see that the energy consumption of all the schemes are non-decreasing functions of ζ, due to the fact that the energy cost of central cloud computing increases with ζ. It is confirmed that the proposed solution outperforms all the baselines, i.e., the energy cost can be significantly reduced. The performance improvement is

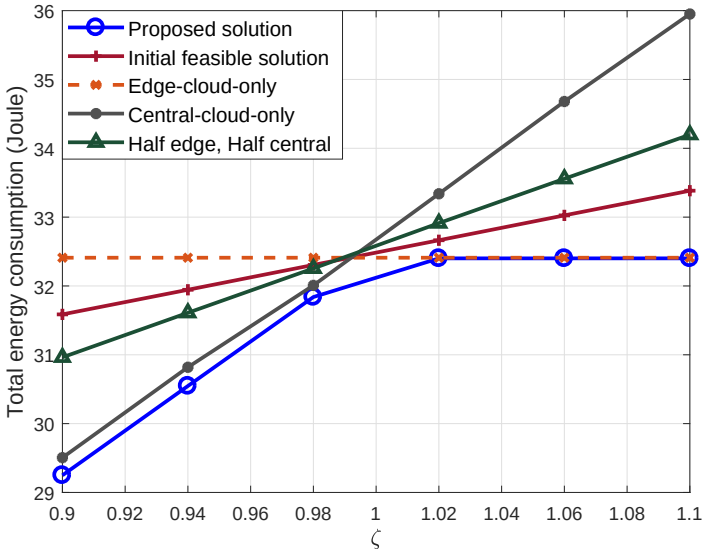


Figure 5.2: The total energy consumption of the system with traditional MIMO backhauls versus the uniform computing energy ratio ζ: $M = 16$, $T_{\text{th}} = 0.3$ s, $\alpha = 0.1$, $I = I_k = 5$ Mbits, $f = f_k = 6$ GHz for $k \in \mathcal{K}$.

particularly noticeable compared with the Edge-cloud-only scheme in the range of $\zeta < 1$, the traditional Central-cloud-only scheme in the range of $\zeta > 1$, and the Half edge, Half central scheme in the whole range of ζ. In addition, the proposed solution also consumes much less energy than the Initial feasible solution, demonstrating the performance enhancement of jointly optimizing the system parameters.

Figure 5.3 depicts the total energy consumption of the system versus the uniform task sizes $I = I_k, k \in \mathcal{K}$ for the cases of $\zeta = 0.9$ and $\zeta = 1.1$. It is easy to understand that computing more input data consumes more energy, and thus the energy cost of each scheme increases with I. Again, we see that the proposed solution is superior to the baseline solutions in all cases. For the case of $\zeta = 0.9$, the performance of the Central-cloud-only solution is very close to the proposed one since the central cloud is dominant in this case, i.e., more UEs tend to use central cloud computing for saving energy. For the case of $\zeta = 1.1$, the advantage of the

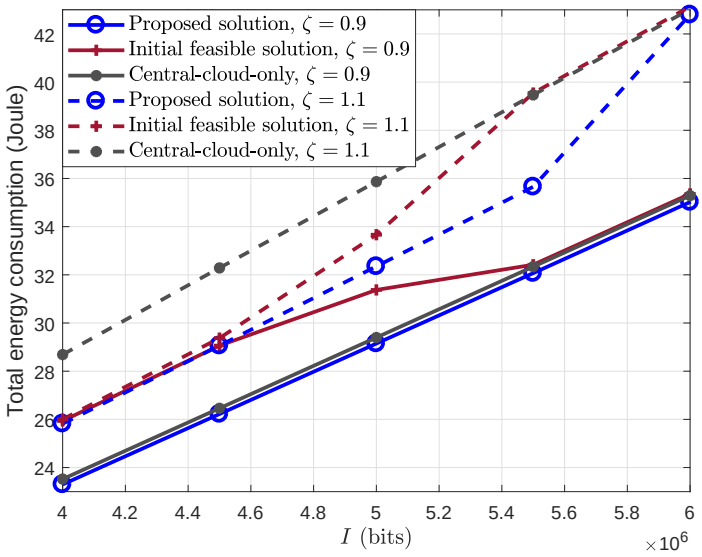


Figure 5.3: The total energy consumption of the system with traditional MIMO backhauls versus the uniform task size I: $M = 16$, $T_{\rm th} = 0.3$ s, $\alpha = 0.1$, $f = f_k = 6$ GHz for $k \in \mathcal{K}$.

proposed scheme becomes more obvious compared with the baselines, and actually this case is more common in practice since the central cloud tends to consume more energy for computing because of the higher CPU frequency. We observe that the results of the proposed solution approach to those of the Central-cloud-only solution when I becomes large, indicating that more UEs tend to select the central cloud for computing, i.e., central cloud computing plays an important role in dealing with relatively large tasks. The reason is that when the task size is large, the edge processing latency constraint C3 of problem (5.12) may be no longer satisfied due to the limited edge computing capability, and central cloud has to be chosen for computation.

Figure 5.4 shows the total energy consumption of the system varying with the latency threshold of edge processing for the cases of $\zeta = 0.9$ and $\zeta = 1.1$. It is seen that the proposed solution is a non-increasing function of $T_{\rm th}$ and outperforms the baselines in both cases. The Central-cloud-only solution is insensitive to $T_{\rm th}$,

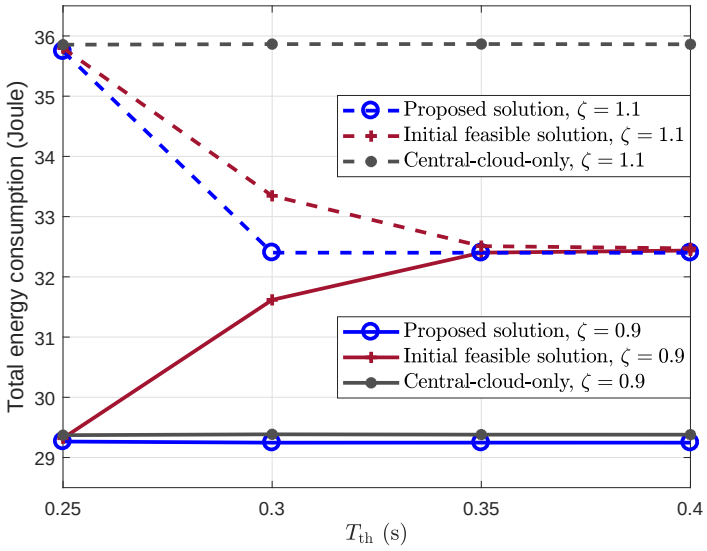


Figure 5.4: The total energy consumption of the system with traditional MIMO backhauls versus the latency threshold of edge processing T_{th}: $M = 16$, $\alpha = 0.1$, $I = I_k = 5$ Mbits, $f = f_k = 6$ GHz for $k \in \mathcal{K}$.

and its performance is almost invariant thanks to its super computing capability for low computing latency. Note that all the solutions consume almost same amount of energy when T_{th} is small, e.g., $T_{\text{th}} = 0.25$ s in this figure. The reason is that the edge processing latency constraint C3 cannot be met and only central cloud computing can be employed to satisfy the latency constraints. For the case of $\zeta = 0.9$, the performance gap between the proposed solution and the Central-cloud-only is small since central cloud computing is dominant, and both solutions perform much better than the Initial feasible solution. It is interesting to note that the the curve of the Initial feasible solution is an increasing function of $T_{\text{th}} \in [0.25, 0.4]$ s when $\zeta = 0.9$. This is because the edge cloud computing becomes more feasible as T_{th} increases, and the initial solution allowing more UEs to choose edge cloud for computing while in fact central cloud computing saves more energy, which indicates the importance of optimizing cloud selection in improving the system performance. For the case of $\zeta = 1.1$, the consumed energy of the proposed solution decreases

with $T_{\rm th}$ since more UEs are allowed to choose the energy-efficient edge cloud computing for large $T_{\rm th}$.

5.5.2 Benefits of Massive MIMO Backhauls

In this subsection, we mainly illustrate the performance of the considered heterogeneous edge/central cloud computing system with massive MIMO backhauls, to confirm the benefits of equipping massive antennas at the MBS in improving the system performance. Here, we focus on MRC and ZF beamforming at the MBS, as studied in Section 5.4.

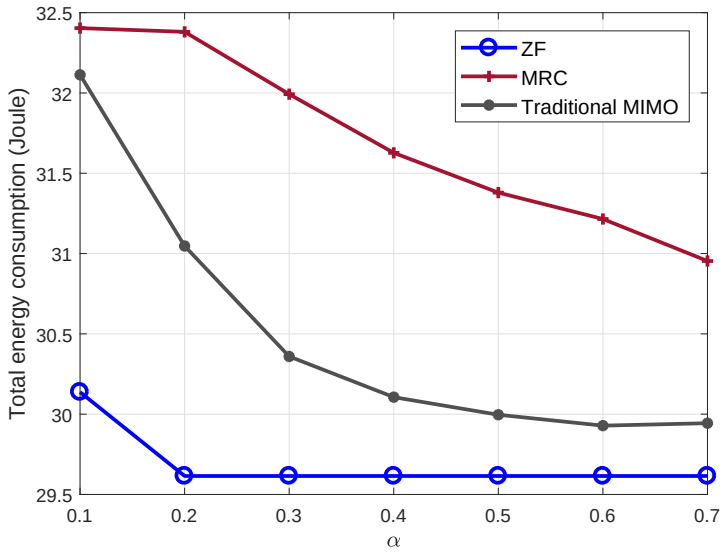


Figure 5.5: The total energy consumption of the system versus the latency ratio parameter α: $M = 128$ for massive MIMO backhauls, $M = 8$ for traditional MIMO backhauls, $T_{\rm th} = 0.3$ s, $\zeta = \zeta_k = 0.9$, $I = I_k = 5$ Mbits, $f = f_k = 6$ GHz for $k \in \mathcal{K}$.

Figure 5.5 and Figure 5.6 depict the total energy consumption and the corresponding percentage of UEs that select edge cloud for computing versus the ratio parameter α, respectively. It is seen from Figure 5.5 that the energy consumption of each scheme decreases with α since less power will be consumed for backhaul

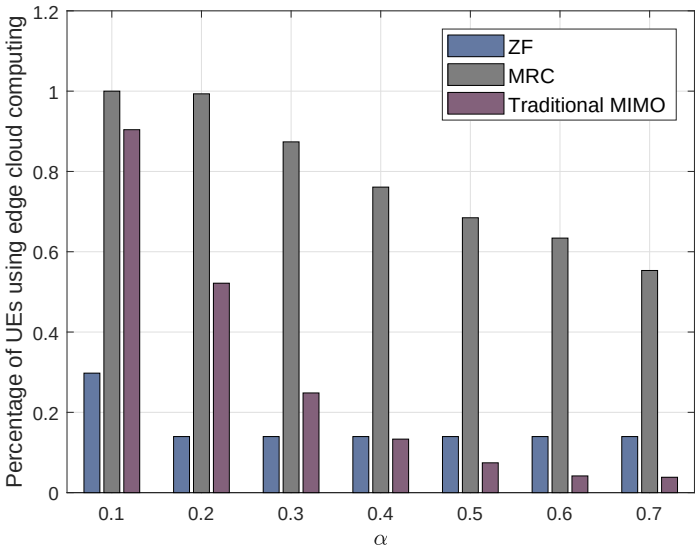


Figure 5.6: The percentage of UEs that select edge cloud computing versus the ratio parameter α: $M = 128$ for massive MIMO backhauls, $M = 8$ for traditional MIMO backhauls, $T_{\rm th} = 0.3$ s, $\zeta = \zeta_k = 0.9$, $I = I_k = 5$ Mbits, $f = f_k = 6$ GHz for $k \in \mathcal{K}$.

transmission with a higher α according to the backhaul latency constraint $\mathrm{C2}$ of problem (5.12). This result is also reflected by Figure 5.6 where the percentage of UEs using edge cloud computing decreases, which means that more UEs choose to use the central cloud for computing as α increases so as to save more energy. Obviously, the energy consumed by the ZF scheme is less than that of the MRC scheme and the solution with traditional MIMO backhauls, which demonstrates the benefits of using ZF beamforming and large antenna arrays at the MBS. Moreover, for the ZF scheme, the percentage of UEs using edge cloud is lower than that of the MRC and traditional MIMO schemes when $\alpha < 0.4$. In contrast, the MRC scheme only uses the edge cloud for computing when $\alpha \leq 0.2$. This is because the backhaul latency constraint $\mathrm{C2}$ in (5.12) for central cloud processing cannot be satisfied with a small α when MRC receiver is adopted at the MBS due to the inter-SBS interference. Based on these two figures, we see that the consumed energy

of the ZF scheme as well as the corresponding percentage of UEs served by edge cloud decrease very slowly, and are almost unchanged for $\alpha \geq 0.2$, which further indicates that the ZF scheme can provide more stable and higher-speed backhaul transmission for computation tasks offloading.

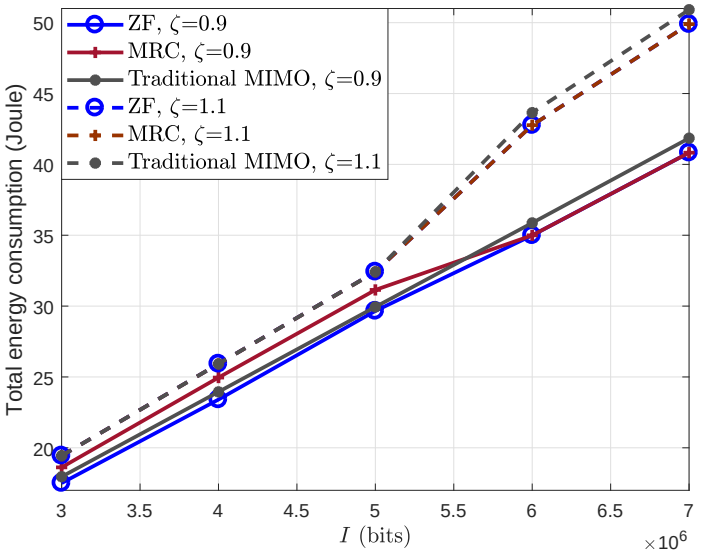
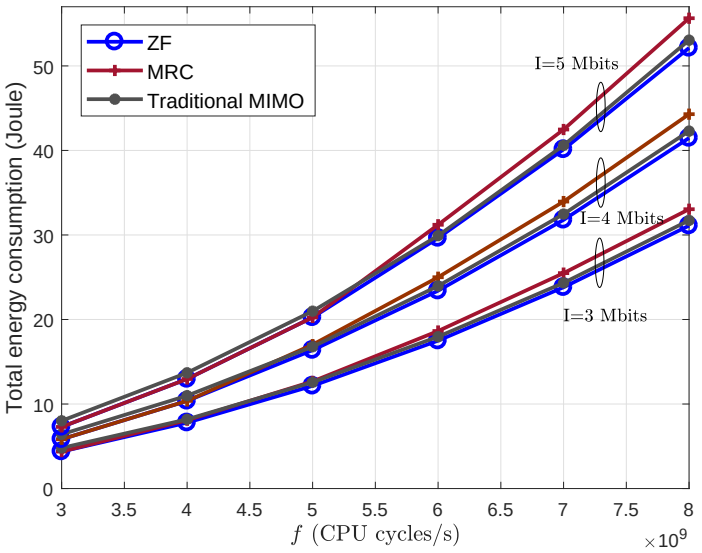


Figure 5.7: The total energy consumption of the system versus the uniform task size I: $M = 128$ for massive MIMO backhauls, $M = 8$ for traditional MIMO backhauls, $T_{\text{th}} = 0.3$ s, $\alpha = 0.6$, $f = f_k = 6$ GHz for $k \in \mathcal{K}$.

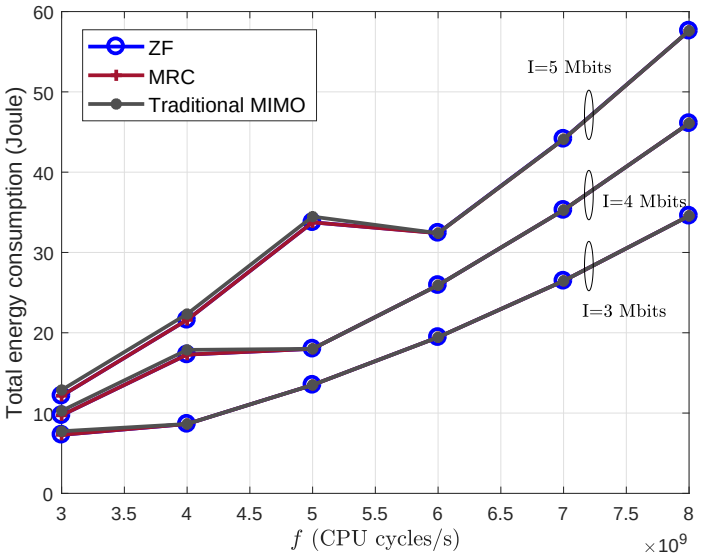
Figure 5.7 shows the total energy consumption of the system versus the uniform task size I for the cases of $\zeta = 0.9$ and $\zeta = 1.1$. Similar to Figure 5.3, all the curves increase with I as expected. The ZF scheme outperforms the MRC scheme and the traditional MIMO scheme. For the case of $\zeta = 0.9$, the ZF scheme and the traditional MIMO scheme are dominated by central cloud computing, while the MRC scheme experiences a gradual transition from edge-cloud-dominant to central-cloud-dominant and more UEs tend to choose the central cloud for computing so as to satisfy the processing latency constraint as well as saving energy. For the case of $\zeta = 1.1$, all the schemes are edge-cloud dominant when $I \leq 5$ Mbits,

and then gradually become central-cloud-dominant as I increases. It is confirmed that the ZF scheme with massive MIMO backhauls has the advantage of handling the computation-intensive tasks.

Figure 5.8(a) and Figure 5.8(b) depict the total energy consumption of the system versus the edge clouds' uniform CPU clock frequency $f = f_k, k \in \mathcal{K}$ in the case of $\zeta = 0.9$ and $\zeta = 1.5$, respectively. According to these two figures, we see that the effect of f is heavily reliant on both the computing task size I and ζ. When I is not large and $\zeta < 1$, the network's energy consumption may increase with f as shown in Figure 5.8(a), where the curves of all the schemes increase with f and the increasing rates become higher when enlarging I. This is due to the fact that when I is not large and $\zeta < 1$, the energy consumption of the central cloud computing plays a dominant role in contributing to the total energy consumption. In this case, the advantage of using ZF scheme becomes more obvious as f grows large. However, when $\zeta > 1$, network's energy consumption may decrease with f in certain scenario as shown in Figure 5.8(b), where there is an obvious decrease as $f \in [5, 6] \times 10^9$ cycles/s(Hz) in the case of $I = 5$ Mbits. The reason is that when f is small, e.g., less than 4×10^9 cycles/s in Figure 5.8(b), the edge processing latency constraint C3 may not be satisfied and central cloud computing becomes the only option. As f increases, edge cloud computing becomes feasible for more UEs to save energy, and the total energy cost will decrease accordingly. In addition, it is seen from Figure 5.8(b) that the energy consumption of the three considered schemes are very close due to the fact the edge cloud computing is dominant for energy saving in this case.



(a) $M = 128$ for massive MIMO backhauls, $M = 8$ for traditional MIMO backhauls, $T_{\text{th}} = 0.3$ s, $\alpha = 0.6$, $\zeta = \zeta_k = 0.9$ for $k \in \mathcal{K}$.



(b) $M = 128$ for massive MIMO backhauls, $M = 8$ for traditional MIMO backhauls, $T_{\text{th}} = 0.3$ s, $\alpha = 0.6$, $\zeta = \zeta_k = 1.5$ for $k \in \mathcal{K}$.

Figure 5.8: The total energy consumption of the system versus SBSs' uniform CPU clock frequency f.

5.6 Summary

In this chapter, we studied the joint design of computing services when edge cloud computing and central cloud computing coexist in a two-tier HetNet with MIMO or massive MIMO self-backhauls. By jointly optimizing the cloud selection, the UEs' transmit powers, the SBSs' receive beamforming vectors and the transmit covariance matrices, the network's total energy consumption for task offloading and computation can be minimized while meeting both the edge processing and central processing (backhaul) latency constraints. An iterative algorithm was proposed to solve the formulated non-convex mixed-integer optimization problem, which can ensure the convergence and that better performance can be achieved than any existing feasible solutions. The numerical results have further confirmed that the proposed solution can greatly enhance the system performance, especially compared with the edge-cloud-only and central-cloud-only computing schemes, indicating the great value of cooperation between the edge and central clouds. Moreover, we showed that the massive MIMO backhauls can largely decrease the complexity of the proposed algorithm while achieving even better performance.

Chapter 6

Conclusions

This dissertation focus on the design and optimization of applying MEC in wireless communication networks. Chapter 2 is the foundation of this thesis, which introduces some fundamental concepts and state-of-the-art works. Then in Chapter 3, Chapter 4 and Chapter 5, we demonstrate the works of design and optimization of MEC in wireless powered cooperation-Assisted systems, UAV-assisted relaying systems, and HetNets with CCC, respectively. Next, we summarize the conclusions and contributions of each chapter in detail.

Chapter 2: Fundamental Concepts and State-of-the-Art Works. In this chapter, we present the fundamental concepts used in this thesis, such as mobile cloud computing, mobile edge computing, wireless power transfer, and UAV-enabled communications, including not only the rationale behind these concepts but also the descendable concepts. Besides, as two important performance metrics for task computing, the basic expressions and derivations related to energy consumption and latency are also shown in this chapter. Moreover, comprehensive literature reviews related to the concepts are given to demonstrate the relevant state-of-the-art

works.

Chapter 3: Mobile Edge Computing in Wireless Powered Cooperation-Assisted Systems. The conclusions and contributions of this chapter are summarized as follows:

- **Wireless Powered MEC Architecture with User Cooperation** — In this chapter, a wireless powered MEC system is studied, in which two mobile devices are first energized by the WPT from an AP and then they can offload part or all of their computation-intensive latency-critical tasks to the AP connected with an MEC server or an edge cloud. This harvest-then-offload protocol operates in an optimized time-division manner. To overcome the double-near-far effect for the farther mobile device in WPCNs, cooperative communications in the form of relaying via the nearer mobile device is considered for offloading.

- **Problem Formulation with Joint Optimization on AP's WPT power, UEs' Offloading Power, and Time Allocation**— Our aim is to minimize the AP's total transmit energy through jointly optimize the AP's energy transmit power, UEs' offloading power, and time allocation, subject to the time allocation constraint, computation task constraints, and energy harvesting causality constraints. We first formulate the AP's transmit energy minimization (APTEM) problem, which is a non-convex optimization problem and difficult to solve directly. We then equivalently transform the APTEM problem into a min-max optimization problem which is also turned out to be equivalent to the AP's transmit power minimization (APTPM) problem.

- **Algorithm Design with A Two-Phase Approach** — The formulated min-

max optimization problem is optimally tackled by a two-phase approach. In the first phase, the inner sum-energy-saving maximization (SESM) problem based on a given energy transmit power is solved by the Lagrangian method, where the optimal offloading decisions with joint power and time allocation are found in closed or semi-closed form. We prove that the optimal offloaded data sizes of the two users have threshold-based structures in relation to some offloading priority indicators. Then in the second phase, a simple bisection search is adopted to obtain the AP's minimum energy transmit power based on the solution of the SESM problem, resulting in the joint-optimal solution. It is shown the proposed algorithm is with low-complexity, at most with the order of $\mathcal{O}(1)\ln(1/\sigma)\ln(1/\delta)$, where $\sigma, \delta > 0$ respectively denote the computational accuracies of two tiers of Bi-section search in the algorithm.

- **Design Insights with Considerable Performance Improvement** — Numerical results verify the theoretical analysis of the proposed cooperative computation offloading scheme, and it demonstrates that the optimized MEC system utilizing cooperation has significant performance improvement over systems without cooperation. It is also shown that the proposed scheme not only achieves significant performance improvement but also demonstrates great effectiveness in handling computation-intensive latency-critical tasks and resisting the double-near-far effect in WPCNs.

- **Practical Implications and Applications for Wireless Powered Cooperation-Assisted MEC Systems** — In this chapter, we leverage the the technology of user cooperation to resist the double-near-far effect rooted in wireless powered MEC systems. It is verified by the simulation results that significant performance improvement can be achieved by the

proposed algorithm compared with other benchmark schemes. More importantly, this work provides fundamental basis and instructive insights for practical implementations of applying wireless powered cooperation-assisted MEC in 5G and beyond networks. The ever growing mobile and IoT devices along with the rapid evolution of 5G communication technologies have given rise of the massive connectivity for fulfilling various novel applications. Even though this massively connected feature bring challenges for stringent requirements of computing and energy resources, it also offers opportunities since massive connectivity can help facilitate the cooperation among user devices. Besides, WPT has been widely regarded as a promising solution to provide sustainable energy supply for the mobile and IoT devices in the practical networks. In conclusion, the architecture of wireless powered cooperation-assisted MEC proposed in this thesis provides a paradigm for providing sustainable energy supply and user-cooperated MEC services in the future 5G and beyond networks with massive connectivity.

Chapter 4: Mobile Edge Computing in UAV-Assisted Relaying Systems. The conclusions and contributions of this chapter are summarized as follows:

- **UAV-Assisted MEC Architecture** — In this chapter, we consider a UAV-assisted MEC architecture with a partial offloading mode where the cellular-connected UAV serves as a mobile computing server as well as a relay to help the UEs complete their computing tasks or further offload their tasks to the AP for computing. This architecture takes full advantage of the UAV's energy-efficient LoS transmissions, and makes proper use of the computing resources at both the UAV and AP through cooperation between each other.

- **Problem Formulation with Joint Computation Resource Scheduling,**

Bandwidth Allocation and UAV's Trajectory Optimization—Our aim is to minimize the weighted sum energy consumption (WSEC) of the UAV and the UEs subject to the UEs' task constraints, the information-causality constraints, the bandwidth allocation constraints and the UAV's trajectory constraints, by jointly optimizing the computation resource scheduling, the bandwidth allocation, and UAV's trajectory iteratively. The formulated problem is complicated and non-convex due to the coupled optimization variables.

- **Alternating Algorithm Design with Guaranteed Convergence** — An alternating optimization algorithm is devised to decouple the optimization variables, through which the formulated problem can be properly solved by addressing three subproblems iteratively. Note that the computation resource scheduling parameters, including the offloading/downloading task sizes and the CPU frequencies at each UE and the UAV, as well as the bandwidth allocation parameters are obtained by leveraging the Lagrange duality method, and that the corresponding Lagrange multipliers associated with the inequality constraints can be obtained using the subgradient method while those associated with the equality constraints can be obtained through bi-section search. The subproblem relating to the UAV's trajectory optimization can be efficiently solved by CVX [128] based on the SCA method. Besides, the convergence of the proposed algorithm can be guaranteed, and the required complexity appears to be acceptable.

- **Design Insights with UAV's Trajectory and Significant Performance Improvement** — Numerical results are presented to show the optimized trajectories of the UAV under different scenarios and the significant perfor-

mance enhancement by leveraging the proposed algorithm when compared to existing schemes, such as the one with a preset UAV trajectory, the scheme with task offloading only, the scheme with equal bandwidth allocation, and the local computing scheme without offloading. Moreover, the proposed algorithm is capable of providing more stable performance in adapting to the changes in the operating environment, and its advantages will become much more prominent when dealing with the computation-intensive and latency-critical tasks.

- **Practical Implications and Applications for UAV-assisted MEC systems** — In this chapter, we resort to the technology of UAV communications to enhance the performance of a MEC system serving multiple ground UEs with a powerful MEC server co-located at the AP. The flexible movement of the assisted UAV brings an additional degree of freedom, and we can observe that significant performance improvement can be achieved by effectively design the UAV's trajectories. In the future communication networks, UAV will play an important role for facilitating various novel communication and computing applications thanks to its highly flexible properties that fixed APs or BSs cannot reach. The UAV-assisted MEC architecture proposed in this thesis can be easily applied in the practical scenario congregated with a large number of users such as the venues of large conferences or expositions, where each UAV can not only act as a moving MEC server providing shared computing resources for UEs but also as a moving relay bringing convenient connections between the AP and UEs. In conclusion, it is of great benefits to explore the UAVs' potentials and their cooperation with cellular-based APs in practical MEC systems, where better communication and computing performance can

be achieved by properly designing the UAV's trajectories with optimized resource allocation according to the requirements of the applications.

Chapter 5: Mobile Edge Computing in Heterogeneous Cellular Networks with Central Cloud Computing. The conclusions and contributions of this chapter are summarized as follows:

- **Hybrid Edge/Central Cloud Computing Architecture** — In this chapter, we consider a hybrid edge and central cloud computing architecture in a two-tier HetNet, including one macro cell with a multi-antenna MBS and multiple small cells each with a multi-antenna SBSs. The edge clouds with limited computing capabilities are co-located at or linked to the SBSs by error-free optical fibers while the central cloud with ultra-high computing capability is connected with the MBS through optical fibers as well. The binary offloading mode is adopted, and thus the UEs can offload their computation tasks directly to the SBSs to access the edge cloud computing services (edge computing mode) or further offload to the MBS through the restricted MIMO/massive MIMO backhauls to utilize the central cloud computing services (central computing mode). Cooperation of edge and central clouds will improve the quality-of-service (QoS) and ensure the scalability and load balancing between the edge and central clouds.

- **Problem Formulation with Joint Optimization on the Cloud Selection, Access Transmit Powers, Receive Beamforming Vectors and Backhaul Transmit Covariance Matrices** — Our aim is to minimize the network's energy consumption for task offloading and computation under both the central and edge processing latency constraints through jointly optimizing the

cloud selection, the UEs' transmit powers, the SBSs' receive beamforming vectors, and the SBSs' transmit covariance matrices. The central processing latency constraints require the backhaul transmission latencies being lower than the corresponding computing latencies at the edge clouds; otherwise, the central cloud will not be selected. The edge processing latency constraints require the corresponding latencies not exceeding a targeted threshold to guarantee the QoS provided by the edge clouds. A mixed-integer and non-convex optimization problem is formulated accordingly, which is NP-hard in general. For the case of massive MIMO backhauls, we consider two low-complexity linear processing methods, namely MRC and ZF, and the corresponding optimization problems can be much simplified.

- **Algorithm Design with MIMO and massive MIMO Backhauls** — An iterative algorithm based on decomposition is developed to solve the combinatorial mixed-integer and non-convex optimization problem corresponding to the case with traditional MIMO backhauls. In particular, we show that in each iteration, the UEs' transmit powers and the SBSs' receive beamforming vectors can be optimized in closed-form, and the SBSs' transmit covariance matrix solution is obtained by leveraging a successive pseudoconvex optimization approach. In addition, the massive MIMO backhaul solutions can be easily obtained thanks to the unique features of massive MIMO transmission, which significantly reduces the complexity of the algorithm. The practicality of the proposed algorithm lies in that it can properly address the issues of cloud selection and resource allocation for a HetNet architecture with hybrid edge/central cloud computing resources while considering the physical properties of wireless backhauls.

- **Design Insights With Performance Improvement and Complexity Reduction** — Numerical results are presented to demonstrate the efficiency of the proposed algorithm and shed light on the effects of key parameters such as the offloaded task size, edge processing latency threshold, and edge clouds' CPU frequency. It is confirmed that the solution of the integrated edge and central cloud computing scheme proposed in this work can achieve better performance than the schemes with edge (cloud) computing alone or central cloud computing alone, and outperforms all the other benchmark solutions. In addition, low-complexity massive MIMO solution with ZF receiver could always outperform the solution with traditional MIMO backhauls, while the solution with MRC receiver could achieve similar or better performance than the traditional MIMO one in certain scenarios.

- **Practical Implications and Applications for Hybrid Edge-Central Cloud Computing Systems** — In this chapter, a practical cloud computing scenario with the coexistence of edge clouds and central cloud is considered in a two-tier heterogeneous cellular network with a macro cell and multiple small cells. The complementary benefits can be achieved through the cooperation between the edge and central clouds by taking into the account of the limitation of wireless backhauls. It is an inexorable trend that both the central clouds and the edge clouds will coexist in the future networks since the edge cloud computing cannot entirely replace the central clouds for completing highly computation-intensive application tasks due to its relatively limited computing capabilities compared with central clouds. Coexisting with central clouds can guarantee the QoS and user experience even in the situations that the computing demands exceed the abilities of the edge clouds. In addition,

the deployment of edge clouds at the SBSs can significantly alleviate the backhaul congestion since a large proportion of computation tasks with small and medium sizes can be completed at the edge clouds without the need of backhaul offloading. Moreover, the advanced technologies of massive MIMO and mmWave can further facilitate the cooperation between the edge and central clouds, achieving better performance with much reduced computational complexity. In a word, the hybrid edge-central cloud computing architecture proposed in this thesis can provide guidelines for the design of the future networks with coexistence of both edge and central cloud computing.

Chapter 7

Future Works

Driven by the motivations discussed in Section 1.2, we completed the research works in this thesis, which addresses the design and optimization of applying MEC in wireless powered cooperation-Assisted systems, UAV-assisted relaying systems, and HetNets with CCC. Actually, our works in this thesis can be further extended to more general or practical scenarios which are regarded as promising research directions for our future works. In this chapter, we will present some potential future works based on this thesis.

7.1 Extensions of MEC in Wireless Powered Cooperation-Assisted Systems

Our work shown in Chapter 3 focuses on the wireless powered cooperation-assisted MEC model for only a three-node scenario, comprising an AP, and two near-far UEs, all with a single antenna. However, extensions to other more complex scenarios are possible, which are also the potential directions of our future works.

This section discusses some straightforward approaches to extend the proposed system in Chapter 3 to more general settings, including the scenarios with multi-antenna AP, more UEs, and computing resource sharing.

7.1.1 Multi-antenna AP

In this case, we consider that the AP is equipped with multiple antennas. Hence, the design of the transmit energy beamforming and the received signal combining at the AP will be handled to improve the network performance giving the multiple antenna capability of the AP. Such a design can be easily achieved by using maximum ratio transmission for wireless power transfer and maximum ratio combining for data reception at the AP. The formulation and approach will be more or less the same except that the resulting channel coefficients after the antenna processing is considered.

7.1.2 More UEs

In Chapter 3, our proposed method in its current form addresses the near-far problem by pairing two UEs (one "near" user and another "far" user) for cooperation. Therefore, for the cases with multiple UEs (far more than two), a natural approach would be to list, then rank and pair users according to their distances from the AP. Communications among different pairs can be dealt with over orthogonal channels within the same cell covered by the AP. By doing so, our proposed solution could be adopted directly. Not allowing different pairs to occupy the same radio channels makes sense because the intra-cell interference would be too much to bear unless advanced interference mitigation techniques are in place. In that case, user pairing has to be done with consideration of the interference levels, as the interference is an

important indicator of the system performance, which will significantly affect the energy consumption at the UEs as well as the AP.

Same goes to extend the proposed work to a multi-cell scenario where inter-cell interference is a crucial factor. After a proper user pairing with consideration of interference control and balancing, our proposed method in Chapter 3 can be directly applied, although the pairing will be more challenging.

7.1.3 Computing Resource Sharing

Another possible extension is to allow users to share not only the radio resources (i.e., power and relaying cooperation as in our current work) but also the computing resources, where the users with stronger computing capacities can help weaker users complete their computation tasks. In this scenario, the required optimization will be much more complex because the energy consumption for carrying out tasks for others and sending back the results to others will need to be evaluated and compared with that for simply relaying the decoded data to the AP. The overall optimization problem can be formulated in a similar manner with the emphasis on minimizing the transmit energy of the AP but the required optimization is not believed to be convex. The exact way to tackle this will require further analysis and it is likely to be considered in our future work.

7.2 MEC in Wireless Powered System with Cooperative UAV

In traditional cellular-based MEC works, the UEs usually resort to the APs for help to complete their offloaded computation tasks, while in the UAV-enabled MEC

architectures, the UEs normally rely on the UAV to handle their offloaded tasks. As mentioned in the Chapter 4, the cooperation between the AP and the UAV is potential and sometimes necessary for completing UEs' tasks due to the facts that the AP can not always provide good connections to some edge users and the size-constrained UAV is resource-limited especially compared with the grid powered AP. In order to make the resource-limited UAV and UEs operate in a sustainable way, the technology of WPT or laser charging can be leveraged to transfer energy from the AP to the UAV and UEs, which is a good way to to fully exploit the AP's abundant grid power supply and further facilitate the cooperation between the UAV and the AP. Based on the analysis above, we plan to construct a wireless powered UAV-assisted MEC architecture, where the UAV cooperates with the AP to compute UEs' offloaded task-input data with sustainable energy supply. This kind of architecture is capable of making full use of both the AP and the UAV's advantages and suppress their disadvantages by leveraging the cooperation between the AP and the UAV, which is a promising research direction that we are now focusing on.

The wireless powered UAV-assisted MEC architecture is shown in Figure 7.1, which consists of an AP, a cellular-connected UAV, and K ground UEs. It is assumed that the UAV and UEs are endowed with wireless energy-harvesting circuits, communication circuits, and computing processors with limited computing capability. In contrast, the grid power supplied AP is equipped with an ultra-high performance processing server, so that it can provide high-speed transmission rate and superb computing capability. Besides, the AP is endowed with a high power energy transmitter and it can transfer energy to the UAV during the task completion time, so as to provide sustainable energy supply for the UAV to support its operations. Part of the UAV's harvested energy will be further broadcast to the

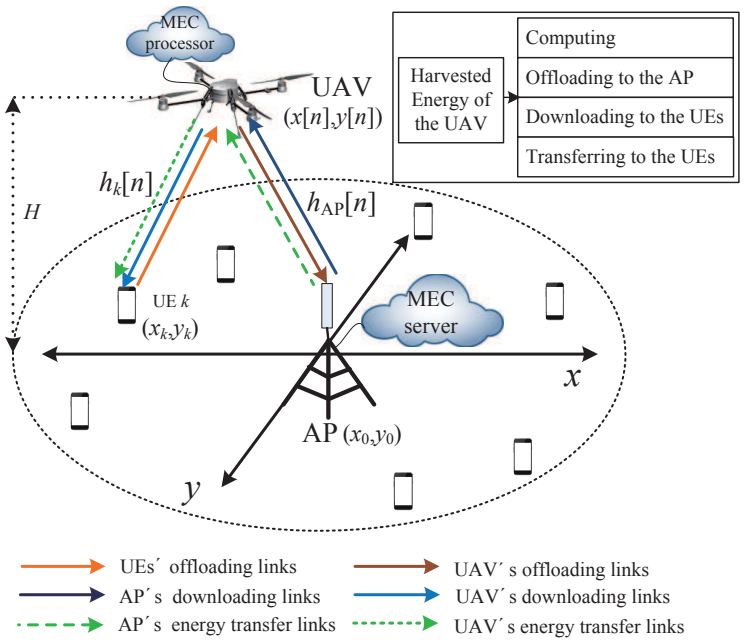


Figure 7.1: An illustration of wireless powered UAV-assisted MEC architecture, where the UAV harvests energy wirelessly from the AP. Besides, the UAV acts as an energy transmitter to offer sustainable wireless energy supply for the UEs, as well as an MEC server and a relay to help the resource-limited UEs compute their offloaded computation tasks or further forward their offloaded tasks to the more powerful processing server at the AP for computing.

UEs, and the remaining part will be utilized for computing and transmissions. We suppose that each UE has a large amount of bit-wise-independent computation task-input data and can be operated in the partial offloading mode. The UAV acts as an MEC server as well as a relay to help the UEs compute their task-input data or further offload their data to the more powerful server at the AP for computing. In this case, it is meaningful to maximize the weighted sum completed task-input bits (WSCTB) of UEs under the task and time allocation, information-causality, energy-causality, and the UAV's trajectory constraints, by jointly optimizing the task and time allocation as well as the UAV's energy transmit power and trajectory. The formulated WSCTB maximization problem should be non-convex due to the strongly coupled optimization parameters, and finding a proper solution is non-trivial. A conference paper [71] has been published based on this architecture, and

we are now focusing on a related journal paper.

7.3 MEC in Cache-Enable Multi-Cell Systems

With the rapid proliferation of mobile devices and Internet-of-things equipment, the global mobile data traffic is growing in an unprecedented way. The explosion of various modern services such as multimedia, smartphone applications, artificial intelligence has driven the demand of wireless communication services shifting from connection-oriented services to content-oriented services. In order to avoid the waste of resources caused by repeatedly transmitting the popular contents, the technology of content caching has been widely regarded as a promising solution [145–149]. Caching the popular contents at the BSs is an effective way for massive content delivery through reducing the distances between popular contents and requesters, and content-caching becomes even more promising considering the gradually reduced prices of storage. Recently, there is a trend of moving the data from cloud to edge [150–154]. In fact, edge caching and edge computing are complementary and can mutually reinforce [155–157].

For one of our future works, we will consider a scenario addressing the edge computing and edge caching simultaneously. As shown in Figure 7.2, a cache-enabled multi-cell MEC architecture is constructed, which comprises N small cells each with a SBS and K UEs. Note that all the SBSs are connected to the corn network through fiber-connected or wireless backhauls. Each UE is assumed to have a hybrid content-aware computation-intensive task, including a computational part and a caching-related part. It is assumed that the users have very limited computing capability, and the computational tasks are atomic and cannot be divided, and thus all the users tend to offload their computation tasks to their associated

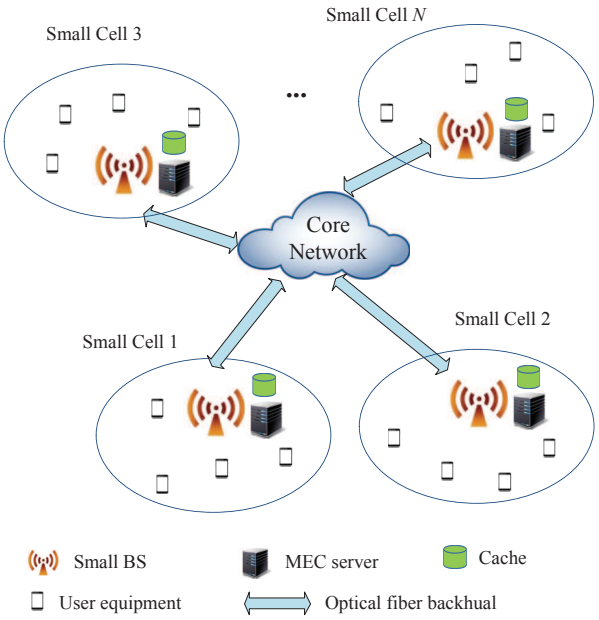


Figure 7.2: An illustration of cache-enabled multi-cell MEC architecture, where N small cells each with a small base station (SBS) to provide caching and computing services to UEs. Each SBS is connected to the corn network through fiber-connected or wireless backhauls.

SBS (MEC server) through uplink transmissions. As for the caching-related part of the task, we assume that all the required contents are saved at the core network, while each SBS has finite caching storage for saving a certain number of contents that is much less than the total amount. Hence, the requested contents of users should either be retrieved directly from the associated SBSs (for contents saved at the corresponding SBSs) or further obtained from the core network (for contents not saved at the corresponding SBSs) through the fiber-connected or wireless backhauls and then send back to the users by the corresponding SBSs.

Based on the assumption above, the users have to complete both uplink communications for computation offloading and downlink communications for content requesting on the premise of satisfying the latency constraints of the tasks. It is assumed that the uplink and downlink communications work in different frequency bands, and the orthogonal multiple access techniques such as TDMA

or OFDMA can be leveraged among users in the same cell. Note that there is no intra-cell interference in each cell, but the inter-cell interference is severe, and should be properly managed so as to achieve satisfactory performance. The uplink and downlink power allocation of UEs, the content placement at the SBSs, the backhaul resource allocation for the SBSs will be considered as the optimization parameters to minimize the total cost, i.e., the energy consumption of the whole system. This optimization problem will be a mixed integer nonlinear programming which is known as a NP-hard problem, and thus solving the problem to obtain a proper solution will be challenging.

Appendices

Appendix A: Proofs in Chapter 3

A.1 Proof of Theorem 3.1

There are two steps to prove **Theorem** 3.1.

1) In order to prove the first result of **Theorem** 3.1, we need the following lemma.

Lemma A.1. *For function $q(z) = e^{(m-1)z} - emz = 0$, there exists a unique root on $z \in (0, \frac{1}{m})$, where $m > 0$ is a constant.*

Proof. Note that $q(0) = 1 > 0$ and $q(\frac{1}{m}) = e(e^{-1/m} - 1) < 0$, indicating that there exists at least one root for $q(z) = 0$ on $z \in (0, 1/m)$. Besides, the second-order derivative of $q(z)$ is non-negative, which means that $q(z)$ is a convex function of z. Hence, we can conclude that there exists one and only one root on $(0, \frac{1}{m})$ for $q(z) = 0$, and it can be easily obtained by a bi-section search on $z \in (0, \frac{1}{m})$. □

We will next show that for the cases of $M_1^+ > 0$ or $\mu_1 \geq (\beta_1 + \beta_2)P_0/z^*$, computation offloading for UE$_1$ is necessary, and thus $\overline{L}_1^* > 0$, $t_1^* > 0$, $q_1^* > 0$. From the two expressions of $\beta_1 \frac{q_1^*}{t_1^*}$ in (3.47) and (3.51), we can get the equation

given below

$$W_0\left(-e^{-(\frac{\eta^*\ln 2}{\lambda_3^* B}+1)}\right) = \frac{-\eta^*\ln 2}{\lambda_3^* B(\beta_1+\beta_2)P_0}. \tag{A.1.1}$$

Denoting $z^* = \frac{\eta^*\ln 2}{\lambda_3^* B} > 0$ and using the definition of the Lambert function, the above equation can be rewritten as

$$e^{\left(\frac{1}{(\beta_1+\beta_2)P_0}-1\right)z^*} - \frac{e}{(\beta_1+\beta_2)P_0}z^* = 0. \tag{A.1.2}$$

Note that $\beta_1\frac{q_1^*}{t_1^*} = \frac{\lambda_3^* B(\beta_1+\beta_2)}{\eta^*\ln 2} - \frac{1}{P_0} = \frac{(\beta_1+\beta_2)}{z^*} - \frac{1}{P_0} > 0$, which means that the above equation should have a unique root z^* on $(0,(\beta_1+\beta_2)P_0)$ because the optimal Lagrange multipliers λ_3^* and η^* are uniquely determined in the convex optimization problem (P3.4). According to **Lemma** A.1, solving (A.1.2) is equivalent to finding the unique root of $q(z)=0$ on $z\in(0,(\beta_1+\beta_2)P_0)$ with $m = 1/(\beta_1+\beta_2)P_0$, and this unique root always exists which can be obtained through a bi-section search on $z^*\in(0,(\beta_1+\beta_2)P_0)$. Therefore, λ_3^* can be expressed by η^* as $\lambda_3^* = \frac{\eta^*\ln 2}{Bz^*}$.

Substituting the expressions of λ_3^*, $(1+\lambda_1^*)$ (in (3.50)) related in η^*, and the definition of β_1 into the condition (3.38) leads to

$$\frac{\partial\mathcal{L}}{\partial\overline{L}_1^*} = \frac{\ln 2}{B}\left(\frac{\mu_1}{(\beta_1+\beta_2)P_0} - \frac{1}{z^*}\right)\eta^*. \tag{A.1.3}$$

Comparison between $\frac{\mu_1}{(\beta_1+\beta_2)P_0}$ and $\frac{1}{z^*}$ according to the result in (3.38) establishes the result of $\overline{L}_1^*$ in (3.53).

Similarly, substituting $\lambda_3^* = \frac{\eta^*\ln 2}{Bz^*}$ into (3.51), the expressions of $\frac{q_1^*}{t_1^*}$ and $\frac{q_{21}^*}{t_{21}^*}$ can be obtained as

$$\frac{q_1^*}{t_1^*} = \frac{1}{\beta_1}\left(\frac{\beta_1+\beta_2}{z^*} - \frac{1}{P_0}\right) > 0, \tag{A.1.4}$$

$$\frac{q_{21}^*}{t_{21}^*} = \frac{1}{\beta_2}\left(\frac{\beta_1+\beta_2}{z^*} - \frac{1}{P_0}\right) > 0. \tag{A.1.5}$$

Based on these, we can further obtain p_1^* and p_{21}^* through the variable revivification, i.e., $p_1^* = \nu_1 g_1 P_0 \frac{q_1^*}{t_1^*}$ and $p_{21}^* = \nu_2 g_2 P_0 \frac{q_{21}^*}{t_{21}^*}$, in combination with $\beta_1 = \frac{\nu_1 g_1 h_1}{N_0}$ and $\beta_2 = \frac{\nu_2 g_2 h_2}{N_0}$, which leads to the results in (3.54) and (3.55).

For the case of $M_1^+ = 0$, $\mu_1 < (\beta_1+\beta_2)P_0/z^*$, it can be derived that $\overline{L}_1^* = 0$ according to condition (3.38), which means that fulfilling UE_1's computation task locally saves more energy, and thus we have $p_1^* = 0$, $p_{21}^* = 0$.

2) Next, we will prove the second result of **Theorem** 3.1. Similarly, we also first show that for the cases of $M_2^+ > 0$ or $\rho(\mu_2) \geq (\beta_1+\beta_2)P_0$, computation offloading for UE_2 is necessary, and thus $L_2^* > 0$, $t_{22}^* > 0$, $q_{22}^* > 0$. According to **Lemma** 3.2, the optimal transmission rate for offloading UE_2's input data, i.e., $\frac{L_2^*}{t_{22}^*}$ can be obtained through the condition (3.35) as

$$\begin{aligned} r_2^* = \frac{L_2^*}{t_{22}^*} &= \frac{B}{\ln 2}\left[W_0\left(\frac{\frac{-h_2\eta^*}{(1+\lambda_2^*)N_0}+1}{-e}\right)+1\right] \\ &\overset{(a)}{=} \frac{B}{\ln 2}\left[W_0\left(\frac{(\beta_1+\beta_2)P_0-1}{e}\right)+1\right] > 0, \end{aligned} \tag{A.1.6}$$

where (a) is obtained through the property of λ_2^* in (3.50) and the definition of β_2. Based on the expression of $g(x)$, its first-order derivative can be expressed as $g'(x) = \frac{N_0 \ln 2}{B} 2^{\frac{x}{B}}$, which is a monotonically increasing function of x. Through the KKT condition (3.39), we can derive that the cases $\frac{\partial \mathcal{L}}{\partial L_2^*}(<,=,>)0$ hold if and only if $\frac{L_2^*}{t_{22}^*}(>,=,<)\frac{B}{\ln 2}\ln \mu_2$, respectively. Hence, the result of L_2^* in (3.56) can be obtained by comparing the expression of $\frac{L_2^*}{t_{22}^*}$ in (A.1.6) and $\frac{B}{\ln 2}\ln \mu_2$, where the definition and property of the Lambert function W_0 [124] should be used. According to (3.21), the optimal transmit power for offloading UE_2's data is $p_{22}^* = \frac{1}{h_2} g\left(\frac{L_2^*}{t_{22}^*}\right)$, giving the

result in (3.57).

For the case of $M_2^+ = 0$, $\rho(\mu_2) < (\beta_1 + \beta_2)P_0$, it can be derived that $L_2^* = 0$ according to (3.39), which means that fulfilling UE_2's task locally saves more energy, thus $p_{22}^* = 0$.

A.2 Proof of Theorem 3.2

Based on the results of **Theorem** 3.1, we can easily derive the expression of t_{22}^* by leveraging the fact of $t_{22}^* = \frac{L_2^*}{r_2^*}$ with the expression of r_2^* in (A.1.6). With the result of t_{22}^*, we can further derive the optimal WPT duration time t_0^* as follows.

For the case of $\overline{L}_1^* = 0$, we understand that $t_1^* = 0$ and $t_{21}^* = 0$, and thus $t_0^* = T - t_{22}^*$. For the case of $\overline{L}_1^* > 0$, combining the results of **Lemma** 3.3, **Lemma** 3.5, and the active time-sharing constraint in (3.24b), establishes the following equation

$$t_1^* + t_{21}^* = \frac{\overline{L}_1^*}{r_{1,1}(\mathbf{p}^*)} = T - t_{22}^* - t_0^*, \tag{A.2.1}$$

which leads to the results in (3.60).

As for the derivation of (t_1^*, t_{21}^*) when $\overline{L}_1^* > 0$, we resort to the results of **Lemma** 3.3 and **Theorem** 3.1, and further derive the following lemma.

Lemma A.2. *The optimal time allocation (t_1^*, t_{21}^*) for cooperatively offloading UE_1's task-input data satisfies*

$$\overline{L}_1^* = L_{1,1}(t_1^*) + L_{1,2}(t_{21}^*) = L_{1,12}(t_1^*). \tag{A.2.2}$$

Proof. According to **Lemma** 3.3 and **Lemma** 3.5, we know that

$$\overline{L}_1^* = (t_1^* + t_{21}^*) r_{1,1}(p_1^*) \leq t_1^* r_{1,12}(p_1^*), \tag{A.2.3}$$

where $\overline{L}_1^*$ and p_1^* have been obtained in **Theorem** 3.1. Since we assume that $h_1 < h_{12}$, then $r_{1,1}(p_1^*) < r_{1,12}(p_1^*)$ holds for sure. With a given feasible P_0 and the corresponding optimal t_0^*, t_{22}^* given above, and p_1^*, p_{21}^*, p_{22}^*, $\overline{L}_1^*$, L_2^* obtained in **Theorem** 3.1, maximizing the SES is equivalent to minimizing the following energy consumption for offloading UE_1's task-input data, i.e.,

$$\begin{aligned} \min_{t_1,t_{21}} \quad & p_1^* t_1 + p_{21}^* t_{21} \\ \text{s.t.} \quad & \text{(A.2.3)},\ t_1 \geq 0,\ t_{21} \geq 0. \end{aligned} \tag{A.2.4}$$

In order to make the cooperative computation offloading strategy effective, we mainly consider the case of $h_1 < h_2$[1], and thus the offloading power satisfies $p_1^* > p_{21}^*$ according to the result of **Theorem** 3.1. If $\overline{L}_1^* = (t_1^* + t_{21}^*) r_{1,1}(p_1^*) < t_1^* r_{1,12}(p_1^*)$ holds, we can always increase t_{21} meanwhile decreasing t_1 with the fixed $t_1 + t_{21} = \overline{L}_1^* / r_{1,1}(p_1^*)$ until $\overline{L}_1^* = (t_1^* + t_{21}^*) r_{1,1}(p_1^*) = t_1^* r_{1,12}(p_1^*)$ holds, which will lead to a smaller objective value of problem (A.2.4). Hence, expression (A.2.2) always holds with the optimal time allocation (t_1^*, t_{21}^*). □

From the result of the above lemma, we can deduce the optimal time division parameters (t_1^*, t_{21}^*) as in (3.61).

A.3 Proof of Lemma 3.6

According to the expression of t_0^* in (3.60), its monotonicity with respect to P_0 is determined by the monotonicity of $\overline{L}_1^* / r_{1,1}(\mathbf{p}^*)$ and $t_{22}^* = L_2^* / r_2^*$ when $\overline{L}_1^* > 0$ or $L_2^* > 0$. From the expression of r_2^* in (A.1.6), it is clear that r_2^* is a monotonic

[1] In this thesis, we mainly consider the case of $h_1 < h_2$, which is most likely to happen based on our assumption that UE_2 is closer to the AP than UE_1 . Actually, if the rare case of $h_1 > h_2$ does happen, we can simply exchange the roles of the two users to apply the proposed scheme, which will achieve similar performance.

increasing function of P_0 due to the fact that the first-branch of Lambert function $W_0(\cdot)$ is a monotonic increasing function. Next, we will prove that P_0/z^* is also a monotonic increasing function of P_0 to further proceed this proof.

From the equation used to obtain z^* in (A.1.2), it is easy to note that z^* is an implicit function of P_0. Besides, equation (A.1.2) can be transformed into another form given by

$$\ln\left(\frac{z^*}{(\beta_1+\beta_2)P_0}\right) = \frac{z^*}{(\beta_1+\beta_2)P_0} - z^* - 1. \tag{A.3.1}$$

As such, the first-order derivative of z^* on P_0 can be found as

$$\frac{\mathrm{d}z^*}{\mathrm{d}P_0} = \frac{z^*\left[(\beta_1+\beta_2)P_0 - z^*\right]}{P_0\left[(\beta_1+\beta_2)P_0 - z^* + (\beta_1+\beta_2)P_0 z^*\right]} \tag{A.3.2}$$

through applying the differentiation rule of the implicit function on the equation (A.3.1). Note that $\frac{\mathrm{d}z^*}{\mathrm{d}P_0} > 0$ always holds since z^* is in the range of $(0, (\beta_1+\beta_2)P_0)$. Thus, the first-order derivative of P_0/z^* can then be expressed as

$$\frac{\mathrm{d}\left(P_0/z^*\right)}{\mathrm{d}P_0} = \frac{(\beta_1+\beta_2)P_0}{(\beta_1+\beta_2)P_0 - z^* + (\beta_1+\beta_2)P_0 z^*}, \tag{A.3.3}$$

which is always positive for $z^* \in (0, (\beta_1+\beta_2)P_0)$. Hence, we can conclude that P_0/z^* monotonically increases with P_0. Then we further prove that $r_{1,1}(\mathbf{p}^*)$ in (3.58) is also a monotonic increasing function of P_0 according to the monotonicity rule of compound function. Note that the thresholds of the offloading decisions for two users in **Theorem** 3.1, i.e., $(\beta_1+\beta_2)P_0/z^*$ and $(\beta_1+\beta_2)P_0$, monotonically increase with P_0, which means that $\overline{L}_1^*$ and L_2^* are two non-increasing piecewise functions of P_0 each with two constant values. Therefore, it is natural that $t_{22}^* =$

L_2^*/r_2^* and $\overline{L}_1^*/r_{1,1}(\mathbf{p}^*)$ are two monotonic decreasing functions of P_0. Therefore, we can conclude that the optimal WPT duration t_0^* in (3.60) is a monotonic increasing function of P_0 for the cases of $\overline{L}_1^* > 0$ or $L_2^* > 0$. When $\overline{L}_1^* = 0$ and $L_2^* = 0$ hold simultaneously, we have $t_1^* = t_{21}^* = t_{22}^* = 0$, and thus t_0^* is fixed as $t_0^* = T$. In conclusion, the WPT duration t_0^* is a monotonic non-decreasing function of P_0.

Appendix B: Proofs in Chapter 4

B.1 Proof of Theorem 4.1

The partial Lagrange function of (P4.1.1) can be expressed as

$$
\begin{aligned}
&\mathcal{L}^{(1)}(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{\rho}, \boldsymbol{\beta}) \\
&= \sum_{k=1}^{K} \Bigg\{ \sum_{n=1}^{N} \Bigg(w_k \Big(E_k^{\mathrm{loc}}[n] + E_k^{\mathrm{off}}[n] \Big) + w_{\mathrm{U}} \Big(E_{\mathrm{U},k}[n] + E_{\mathrm{U},k}^{\mathrm{off}}[n] + E_{\mathrm{U},k}^{\mathrm{down}}[n] \Big) \Bigg) \\
&+ \Bigg(\sum_{n=2}^{N-1} \widetilde{\lambda}_{k,n} \Bigg(\frac{\delta f_{\mathrm{U},k}[n]}{C_k} + L_{\mathrm{U},k}^{\mathrm{off}}[n] \Bigg) - \sum_{n=1}^{N-2} \widehat{\lambda}_{k,n} L_k^{\mathrm{off}}[n] \Bigg) \\
&+ \Bigg(\sum_{n=3}^{N} \widetilde{\mu}_{k,n} L_{\mathrm{U},k}^{\mathrm{down}}[n] - O_k \sum_{n=2}^{N-1} \widehat{\mu}_{k,n} \Bigg(\frac{\delta f_{\mathrm{U},k}[n]}{C_k} + L_{\mathrm{U},k}^{\mathrm{off}}[n] \Bigg) \Bigg) \\
&+ \eta_k \Bigg(\sum_{n=1}^{N-2} L_k^{\mathrm{off}}[n] - \sum_{n=2}^{N-1} \Bigg(\frac{\delta f_{\mathrm{U},k}[n]}{C_k} + L_{\mathrm{U},k}^{\mathrm{off}}[n] \Bigg) \Bigg) \\
&+ \rho_k \Bigg(O_k \sum_{n=2}^{N-1} \Bigg(\frac{\delta f_{\mathrm{U},k}[n]}{C_k} + L_{\mathrm{U},k}^{\mathrm{off}}[n] \Bigg) - \sum_{n=3}^{N} L_{\mathrm{U},k}^{\mathrm{down}}[n] \Bigg) \\
&+ \beta_k \Bigg(I_k - \sum_{n=1}^{N-2} L_k^{\mathrm{off}}[n] - \sum_{n=1}^{N} \frac{\tau}{C_k} f_k[n] \Bigg) \Bigg\},
\end{aligned}
\tag{B.1.1}
$$

where $\boldsymbol{\lambda} = \{\lambda_{k,n}\}_{k\in\mathcal{K},n\in\mathcal{N}}$, $\boldsymbol{\mu} = \{\mu_{k,n}\}_{k\in\mathcal{K},n\in\mathcal{N}}$, $\boldsymbol{\eta} = \{\eta_k\}_{k\in\mathcal{K}}$, $\boldsymbol{\rho} = \{\rho_k\}_{k\in\mathcal{K}}$, $\boldsymbol{\beta} = \{\beta_k\}_{k\in\mathcal{K}}$, $\widetilde{\lambda}_{k,n} = \sum_{i=n}^{N-1} \lambda_{k,i}$, $\widehat{\lambda}_{k,n} = \sum_{i=n+1}^{N-1} \lambda_{k,i}$, $\widetilde{\mu}_{k,n} = \sum_{i=n}^{N} \mu_{k,i}$, and

$\widehat{\mu}_{k,n} = \sum_{i=n+1}^{N} \mu_{k,i}$. The Lagrangian dual function of problem (P4.1.1) can be presented as

$$d^{(1)}(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{\rho}, \boldsymbol{\beta}) = \min_{\mathbf{z}} \ \mathcal{L}^{(1)}(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{\rho}, \boldsymbol{\beta}) \quad \text{s.t. (4.17h)} - \text{(4.17l)}. \tag{B.1.2}$$

Hence, the solution of $\mathbf{z}$ with given dual variables $\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{\rho}, \boldsymbol{\beta}$ can be obtained by solving problem (B.1.2). If the given dual variables are optimal, denoted as $\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{\eta}^*, \boldsymbol{\rho}^*, \boldsymbol{\beta}^*$, then the corresponding solutions are optimal, i.e., $\mathbf{z}^*$. According to the structures of the Lagrange function $\mathcal{L}^{(1)}(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{\rho}, \boldsymbol{\beta})$ and the constraints (4.17h)-(4.17l), it is noted that the problem (B.1.2) can be equivalently divided into K subproblems w.r.t. each UE $k \in \mathcal{K}$ to facilitate parallel execution. Apply the Karush-Kuhn-Tucker (KKT) conditions [123] and let the derivations of $\mathcal{L}^{(1)}(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{\rho}, \boldsymbol{\beta})$ w.r.t. $f_k[n]$, $L_k^{\text{off}}[n]$, $f_{\text{U},k}[n]$, $L_{\text{U},k}^{\text{off}}[n]$, $L_{\text{U},k}^{\text{down}}[n]$ equal to zero, we can thus obtain the corresponding optimal solution given in **Theorem** 4.1 with some straightforward calculations.

B.2 Proof of Lemma 4.1

The optimal Lagrange multipliers $\boldsymbol{\lambda}^*$, $\boldsymbol{\mu}^*$, $\boldsymbol{\eta}^*$, $\boldsymbol{\rho}^*$ and $\boldsymbol{\beta}^*$ related to the optimal solution of problem (P4.1.1) can be obtained by solving the dual problem of (P4.1.1), which is expressed as

$$\max_{\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\eta},\boldsymbol{\rho},\boldsymbol{\beta}} \ d^{(1)}(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{\rho}, \boldsymbol{\beta}) \tag{B.2.1}$$

$$\text{s.t.} \quad \lambda_{k,n} \geq 0, \ \mu_{k,n} \geq 0, \ k \in \mathcal{K}, n \in \mathcal{N}, \tag{B.2.2}$$

$$\sum_{n=1}^{N-2} L_{k,j}^{\text{off}*}[n] = \sum_{n=2}^{N-1} \left(\frac{\delta f_{\text{U},k,j}^{*}[n]}{C_k} + L_{\text{U},k,j}^{\text{off}*}[n] \right),\ k \in \mathcal{K}, \tag{B.2.3}$$

$$O_k \sum_{n=2}^{N-1} \left(\frac{\delta f_{\text{U},k,j}^{*}[n]}{C_k} + L_{\text{U},k,j}^{\text{off}*}[n] \right) = \sum_{n=3}^{N} L_{\text{U},k,j}^{\text{down}*}[n],\ k \in \mathcal{K}, \tag{B.2.4}$$

$$\sum_{n=1}^{N} \frac{\tau}{C_k} f_{k,j}^{\text{off}*}[n] + \sum_{n=1}^{N-2} L_{k,j}^{*}[n] = I_k,\ k \in \mathcal{K}, \tag{B.2.5}$$

where (B.2.3)–(B.2.5) are given to make sure that the obtained Lagrange multipliers $\boldsymbol{\eta}^*$, $\boldsymbol{\rho}^*$ and $\boldsymbol{\beta}^*$ based on the $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$ can make the optimal solution of problem (P4.1.1) satisfy the equality constraints (4.17d)-(4.17f). The optimal $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$ associated with the inequality constraints (4.17b)–(4.17c) can be obtained through the subgradient-based algorithm, which gives the results of $\boldsymbol{\lambda}_{j+1}$ and $\boldsymbol{\mu}_{j+1}$ at the $(j+1)$-th iteration as shown in **Lemma** 4.1.

B.3 Proof of Lemma 4.2

With the achieved $\boldsymbol{\lambda}_{j+1}$ and $\boldsymbol{\mu}_{j+1}$ in **Lemma** 4.1, we can then obtain the $\boldsymbol{\eta}_{j+1}$, $\boldsymbol{\rho}_{j+1}$ and $\boldsymbol{\beta}_{j+1}$ correspondingly. According to the expressions of the optimal solution in **Theorem** 4.1 and the equality constraints in (4.17d)–(4.17f), we can express the value of $\sum_{n=1}^{N-2} L_{k,j+1}^{\text{off}*}[n]$ in the following forms in (B.3.1)–(B.3.4)

$$\sum_{n=1}^{N-2} L_{k,j+1}^{\text{off}*}[n] = I_k - \frac{T}{C_k} \sqrt{\frac{\beta_{k,j+1}}{3 C_k w_k \kappa_k}} \tag{B.3.1}$$

$$= \delta \sum_{n=1}^{N-2} B_k^{\text{off}}[n] \left[\varphi_k^{\text{off}}[n] + \log_2 \left[\widehat{\lambda}_{k,n,j+1} + \beta_{k,j+1} - \eta_{k,j+1} \right]^{+} \right]^{+} \tag{B.3.2}$$

$$= \frac{\delta}{O_k} \sum_{n=3}^{N} B_{\text{U},k}^{\text{down}}[n] \left[\varphi_{\text{U},k}^{\text{down}}[n] + \log_2 \left[\rho_{k,j+1} - \widetilde{\mu}_{k,n,j+1} \right]^{+} \right]^{+} \tag{B.3.3}$$

$$= \sum_{n=2}^{N-1} \left\{ \frac{\delta}{C_k} \sqrt{\frac{[\eta_{k,j+1} - O_k \rho_{k,j+1} + O_k \widehat{\mu}_{k,n,j+1} - \widetilde{\lambda}_{k,n,j+1}]^{+}}{3 C_k w_{\text{U}} \kappa_{\text{U}}}} \right.$$

$$+\delta B_{\mathrm{U},k}^{\mathrm{off}}[n]\Bigg[\varphi_{\mathrm{U},k}^{\mathrm{off}}[n]+\log_2\Big[\eta_{k,j+1}-O_k\rho_{k,j+1}$$
$$+O_k\widehat{\mu}_{k,n,j+1}-\widetilde{\lambda}_{k,n,j+1}\Big]^+\Big]^+\Bigg\}, \tag{B.3.4}$$

where $\widetilde{\lambda}_{k,n,j+1}$, $\widehat{\lambda}_{k,n,j+1}$, $\widetilde{\mu}_{k,n,j+1}$, and $\widehat{\mu}_{k,n,j+1}$ are defined similar to $\widetilde{\lambda}_{k,n}$, $\widehat{\lambda}_{k,n}$, $\widetilde{\mu}_{k,n}$, and $\widehat{\mu}_{k,n}$ in Appendix B.1. The expression (B.3.1) is obtained from (4.17f), (B.3.2) comes from the expression of $\{L_{k,j+1}^{\mathrm{off}*}[n]\}$, (B.3.3) is derived from (4.17d) and (4.17e) with equation $\sum_{n=1}^{N-2} L_{k,j+1}^{\mathrm{off}*}[n] = \frac{1}{O_k}\sum_{n=3}^{N} L_{\mathrm{U},k,j+1}^{\mathrm{down}*}[n]$, and (B.3.4) is obtained from (4.17d).

According to (B.3.1) and the facts that $\sum_{n=1}^{N-2} L_{k,j+1}^{\mathrm{off}}[n] \in [0, I_k]$, $f_k^*[n] \geq 0$, we can derive the range of $\beta_{k,j+1} \in [0, \beta_{k,\max})$ with $\beta_{k,\max} = 3C_k w_k \kappa_k (\frac{I_k C_k}{T})^2$ for $k \in \mathcal{K}$. It is observed from (B.3.1)–(B.3.3) that $\eta_{k,j+1}$ and $\rho_{k,j+1}$ are respectively monotonic non-decreasing and non-increasing implicit functions of $\beta_{k,j+1}$, which further shows that (B.3.4) is also a monotonic non-decreasing function of $\beta_{k,j+1}$. Hence, with the obtained $\boldsymbol{\lambda}_{j+1}$ and $\boldsymbol{\mu}_{j+1}$, and a given $\beta_{k,j+1} \in [0, \beta_{k,\max})$, we can derive the corresponding $\eta_{k,j+1}$ and $\rho_{k,j+1}$ from the equations constituted by (B.3.1) in company with (B.3.2) and (B.3.3), respectively, also using the bi-section search method with the ranges of $\eta_{k,j+1} \in [\eta_{k,j+1}^{\mathrm{low}}, \eta_{k,j+1}^{\mathrm{up}}]$ and $\rho_{k,j+1} \in [\rho_{k,j+1}^{\mathrm{low}}, \rho_{k,j+1}^{\mathrm{up}}]$, where

$$\eta_{k,j+1}^{\mathrm{low}} = \widehat{\lambda}_{k,N-2,j+1} - 2^{\frac{I_k/\delta - \sum_{n=1}^{N-2} B_k^{\mathrm{off}}[n]\varphi_k^{\mathrm{off}}[n]}{\sum_{n=1}^{N-2} B_k^{\mathrm{off}}[n]}}, \tag{B.3.5}$$

$$\eta_{k,j+1}^{\mathrm{up}} = \widehat{\lambda}_{k,1,j+1} + \beta_{k,\max}, \tag{B.3.6}$$

$$\rho_{k,j+1}^{\mathrm{low}} = \widetilde{\mu}_{k,N,j+1}, \tag{B.3.7}$$

$$\rho_{k,j+1}^{\mathrm{up}} = \widetilde{\mu}_{k,3,j+1} + 2^{\frac{I_k O_k/\delta - \sum_{n=3}^{N} B_{\mathrm{U},k}^{\mathrm{down}}[n]\varphi_{\mathrm{U},k}^{\mathrm{down}}[n]}{\sum_{n=3}^{N} B_{\mathrm{U},k}^{\mathrm{down}}[n]}}, \tag{B.3.8}$$

which are obtained from (B.3.2) and (B.3.3) in combination with the definitions of $\widehat{\lambda}_{k,n,j+1}$ and $\widetilde{\mu}_{k,n,j+1}$, and the range of $\beta_{k,j+1}$. The optimal $\beta_{k,j+1}$ and the corresponding $\eta_{k,j+1}$, $\rho_{k,j+1}$ should make the equation formed by (B.3.1) and (B.3.4) satisfied, which indicates the termination of the bi-section search of $\beta_{k,j+1}$, $k \in \mathcal{K}$.

B.4 Proof of Theorem 4.2

The partial Lagrange function of (P4.1.2) is defined as

$$\begin{aligned}
&\mathcal{L}^{(2)}(\mathbf{B}, \boldsymbol{\nu}) = \\
&\sum_{k=1}^{K}\sum_{n=1}^{N}\left(w_k E_k^{\text{off}}[n] + w_{\text{U}}\Big(E_{\text{U},k}^{\text{off}}[n] + E_{\text{U},k}^{\text{down}}[n]\Big)\right) + \\
&\sum_{k=1}^{K}\sum_{n=1}^{N}\nu_{k,n}\big(B_k^{\text{off}}[n] + B_{\text{U},k}^{\text{off}}[n] + B_{\text{U},k}^{\text{down}}[n] - B\big),
\end{aligned} \tag{B.4.1}$$

where $\boldsymbol{\nu} = \{\nu_{k,n}\}_{k\in\mathcal{K},n\in\mathcal{N}}$. The Lagrangian dual function of problem (P4.1.2) can be presented as

$$\begin{aligned}
d^{(2)}(\boldsymbol{\nu}) = \min_{\mathbf{B}}\ & \mathcal{L}^{(2)}(\mathbf{B}, \boldsymbol{\nu}) \\
\text{s.t.}\ & \text{(4.17m)} - \text{(4.17o)}.
\end{aligned} \tag{B.4.2}$$

Hence, the optimal solution of $\mathbf{B}$ with optimal dual variables $\boldsymbol{\nu}^*$ can be obtained by solving (B.4.2). This problem can also be equivalently divided into K subproblems w.r.t. each UE $k \in \mathcal{K}$ to facilitate parallel execution. It is easy to note that the expressions of $E_k^{\text{off}}[n]$, $E_{\text{U},k}^{\text{off}}[n]$ and $E_{\text{U},k}^{\text{down}}[n]$ have similar structures w.r.t. $B_k^{\text{off}}[n]$, $B_{\text{U},k}^{\text{off}}[n]$ and $B_{\text{U},k}^{\text{down}}[n]$, and thus the optimal solution of $B_k^{\text{off}}[n]$, $B_{\text{U},k}^{\text{off}}[n]$ and $B_{\text{U},k}^{\text{down}}[n]$ should have similar structures according to problem (B.4.2). Next, we

will take $B_k^{\text{off}}[n]$ as an example to obtain its closed-form optimal solution versus $\nu_{k,n}^*$ for $k \in \mathcal{K}, n \in \mathcal{N}$. Applying the KKT conditions [123] leads to the following necessary and sufficient condition of $B_k^{\text{off}*}[n]$:

$$\frac{\partial \mathcal{L}^{(2)}(\mathbf{B}, \boldsymbol{\nu})}{\partial B_k^{\text{off}*}[n]} = \nu_{k,n}^* - \frac{L_k^{\text{off}}[n] w_k N_0 \ln 2}{(B_k^{\text{off}*}[n])^2 h_k[n]} 2^{\frac{L_k^{\text{off}}[n]}{B_k^{\text{off}*}[n]\delta}} = 0, \tag{B.4.3}$$

where the optimal dual variable $\nu_{k,n}^*$ should make sure that the equality constraint $B_k^{\text{off}*}[n] + B_{\text{U},k}^{\text{off}*}[n] + B_{\text{U},k}^{\text{down}*}[n] = B$ is satisfied. It is not easy to obtain the closed-form solution of $B_k^{\text{off}*}[n]$ through (B.4.3) directly. By defining $\xi = \frac{L_k^{\text{off}}[n]}{B_k^{\text{off}*}[n]\delta}$, the equation in (B.4.3) can be re-expressed as

$$\xi^2 2^\xi = \frac{\nu_{k,n}^* h_k[n] L_k^{\text{off}}[n]}{\delta^2 w_k N_0 \ln 2} \triangleq \Gamma. \tag{B.4.4}$$

By applying the natural logarithm at the both sides of (B.4.4) leads to

$$\ln \xi + \frac{\ln 2}{2} \xi = \ln \Gamma^{\frac{1}{2}}. \tag{B.4.5}$$

Then applying the exponential operation at both sides of (B.4.5), we can obtain that

$$\frac{\ln 2}{2} \xi e^{\frac{\ln 2}{2}\xi} = \frac{\ln 2}{2} \Gamma^{\frac{1}{2}}, \tag{B.4.6}$$

where e is the base of the natural logarithm. According to the definition and property of Lambert function [124], we have $\frac{\ln 2}{2}\xi = W_0(\frac{\ln 2}{2}\Gamma^{\frac{1}{2}})$, and finally we can express $B_k^{\text{off}*}[n]$ as

$$B_k^{\text{off}*}[n] = \frac{\frac{\ln 2}{2} L_k^{\text{off}}[n]}{\delta W_0\big[\frac{\ln 2}{2}(\frac{\phi_{k,n}}{w_k} h_k[n] L_k^{\text{off}}[n])^{\frac{1}{2}}\big]}, \; n \in \mathcal{N}_1. \tag{B.4.7}$$

Integrating with the cases $B_k^{\text{off}*}[N-1] = B_k^{\text{off}*}[N] = 0$, the complete solution of $B_k^{\text{off}^*}[n]$ in (4.32) can be obtained. The solution of $B_{\text{U},k}^{\text{off}*}[n]$ and $B_{\text{U},k}^{\text{down}*}[n]$ in (4.33) and (4.34) can be obtained in a similar way.

Appendix C: Proofs in Chapter 5

C.1 Proof of Lemma 5.2

The Lagrange function of problem (5.20) is

$$\begin{aligned}\mathcal{L}\left(\mathbf{p}^{\text{u}}, \mathbf{t}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}\right) = &\sum_{k=1}^{K} I_k t_k + \sum_{k=1}^{K} \lambda_k \left(p_k^{\text{u}} - t_k R_k^{\text{a}}(\mathbf{p}^{\text{u}}, \mathbf{w}_k)\right) \\ &+ \sum_{k=1}^{K} \mu_k \left(\tau_k - \gamma_k^{\text{a}}(\mathbf{p}^{\text{u}}, \mathbf{w}_k)\right) \\ &+ \sum_{k=1}^{K} \nu_k \left(p_k^{\text{u}} - P_{\max}^{\text{u}}\right), \end{aligned} \tag{C.1.1}$$

where $\{\lambda_k, \mu_k, \nu_k\}_{k=1}^{K}$ are non-negative Lagrange multipliers. Based on the definition of KKT conditions, we have

$$\frac{\partial \mathcal{L}}{\partial p_k^{\text{u}}} = \lambda_k - \lambda_k t_k \frac{\partial R_k^{\text{a}}}{\partial p_k^{\text{u}}} - \mu_k \frac{\partial \gamma_k^{\text{a}}}{\partial p_k^{\text{u}}} + \nu_k - \sum_{j=1, j\neq k}^{K} \lambda_j t_j \frac{\partial R_j^{\text{a}}}{\partial p_k^{\text{u}}} - \sum_{j=1, j\neq k}^{K} \mu_j \frac{\partial \gamma_j^{\text{a}}}{\partial p_k^{\text{u}}} = 0, \tag{C.1.2}$$

$$\frac{\partial \mathcal{L}}{\partial t_k} = I_k - \lambda_k R_k^{\text{a}} = 0, \tag{C.1.3}$$

$$\lambda_k \left(p_k^{\text{u}} - t_k R_k^{\text{a}}\right) = 0, \tag{C.1.4}$$

$$\mu_k \left(\tau_k - \gamma_k^{\text{a}}\right) = 0, \tag{C.1.5}$$

$$\nu_k \left(p_k^{\text{u}} - P_{\max}^{\text{u}}\right) = 0. \tag{C.1.6}$$

In (C.1.2), we have $\frac{\partial R_j^{\mathrm{a}}}{\partial p_k^{\mathrm{u}}} = -\frac{B^{\mathrm{a}}}{\ln 2}\frac{\left(\gamma_j^{\mathrm{a}}\right)^2|\mathbf{w}_j^H\mathbf{h}_{k,j}^{\mathrm{a}}|^2}{p_j^{\mathrm{u}}|\mathbf{w}_j^H\mathbf{h}_{j,j}^{\mathrm{a}}|^2\left(1+\gamma_j^{\mathrm{a}}\right)}$, and $\frac{\partial \gamma_j^{\mathrm{a}}}{\partial p_k^{\mathrm{u}}} = -\frac{\left(\gamma_j^{\mathrm{a}}\right)^2|\mathbf{w}_j^H\mathbf{h}_{k,j}^{\mathrm{a}}|^2}{p_j^{\mathrm{u}}|\mathbf{w}_j^H\mathbf{h}_{j,j}^{\mathrm{a}}|^2}$. In addition, since $R_k^{\mathrm{a}} > 0$, we have $\lambda_k = \frac{I_k}{R_k^{\mathrm{a}}} > 0$ based on (C.1.3), and then $t_k = \frac{p_k^{\mathrm{u}}}{R_k^{\mathrm{a}}}$ based on (C.1.4). Through (C.1.2)–(C.1.6), we observe that problem (5.20) has the same KKT conditions with the K subproblems shown in (5.21). Likewise, by considering the KKT conditions of K subproblems in (5.21), we find that they are identical to those shown in (C.1.2)–(C.1.6). In other words, problems (5.20) and (5.21) have the same optimal solution.

C.2 Proof of Theorem 5.1

Based on (C.1.2), (C.1.5) and (C.1.6) of Appendix C.1, KKT conditions for subproblem (5.21) is given by

$$\lambda_k + M_k - \frac{B_{\mathrm{a}}}{\ln 2}\frac{\lambda_k t_k \Lambda_k}{1+\gamma_k^{\mathrm{a}}} - \mu_k\Lambda_k + \nu_k = 0, \tag{C.2.1}$$

$$\mu_k\left(\tau_k - \gamma_k^{\mathrm{a}}\right) = 0, \tag{C.2.2}$$

$$\nu_k\left(p_k^{\mathrm{u}} - P_{\max}^{\mathrm{u}}\right) = 0, \tag{C.2.3}$$

where $\Lambda_k = \frac{|\mathbf{w}_k^H\mathbf{h}_{k,k}^{\mathrm{a}}|^2}{\sum_{i=1,i\neq k}^{K} p_i^{\mathrm{u}}|\mathbf{w}_k^H\mathbf{h}_{i,k}^{\mathrm{a}}|^2 + |\mathbf{w}_k^H\mathbf{n}_k|^2}$. From (C.2.1) and the definition of $\gamma_k^{\mathrm{a}} = p_k^{\mathrm{u}}\Lambda_k$ in (5.2), we see that the optimal $p_k^{\mathrm{u}*}$ meets

$$p_k^{\mathrm{u}*} = \frac{B_{\mathrm{a}}}{\ln 2}\frac{\lambda_k t_k}{\lambda_k + M_k - \mu_k^*\Lambda_k + \nu_k^*} - \frac{1}{\Lambda_k}, \tag{C.2.4}$$

where μ_k^* and ν_k^* satisfy the KKT conditions (C.2.2) and (C.2.3), respectively. To explicitly obtain $\{p_k^{\mathrm{u}*}, \mu_k^*, \nu_k^*\}$, we need to consider the following cases:

- Case 1: When $p_k^{\mathrm{u}*} \in \left(\frac{\tau_k}{\Lambda_k}, P_{\max}^{\mathrm{u}}\right)$, $\mu_k^* = \nu_k^* = 0$ according to (C.2.2) and (C.2.3). In this case, $p_k^{\mathrm{u}*} = G_k$ with $G_k = \frac{B_{\mathrm{a}}}{\ln 2}\frac{\lambda_k t_k}{\lambda_k + M_k} - \frac{1}{\Lambda_k}$ according to

(C.2.4). Therefore, if $G_k \in \left[\frac{\tau_k}{\Lambda_k}, P_{\max}^{\mathrm{u}}\right]$, $p_k^{\mathrm{u}*} = G_k$ and $\mu_k^* = \nu_k^* = 0$.

- Case 2: If $G_k < \frac{\tau_k}{\Lambda_k}$, it is seen from (C.2.4) that $\mu_k^* > 0$. In this case, $p_k^{\mathrm{u}*} = \frac{\tau_k}{\Lambda_k}$ and $\nu_k^* = 0$ according to (C.2.2) and (C.2.3). Substituting $p_k^{\mathrm{u}*} = \frac{\tau_k}{\Lambda_k}$ and $\nu_k^* = 0$ into (C.2.4), we obtain $\mu_k^* = \frac{\lambda_k + M_k}{\Lambda_k} - \frac{B_{\mathrm{a}}}{\ln 2}\frac{\lambda_k t_k}{\tau_k + 1}$

- Case 3: If $G_k > P_{\max}^{\mathrm{u}}$, it is seen from (C.2.4) that $\nu_k^* > 0$. In this case, $p_k^{\mathrm{u}*} = P_{\max}^{\mathrm{u}}$ and $\mu_k^* = 0$ according to (C.2.3) and (C.2.2). Substituting $p_k^{\mathrm{u}*} = P_{\max}^{\mathrm{u}}$ and $\mu_k^* = 0$ into (C.2.4), we obtain $\nu_k^* = \frac{B_{\mathrm{a}}}{\ln 2}\frac{\lambda_k t_k}{P_{\max}^{\mathrm{u}} + 1/\Lambda_k} - \lambda_k - M_k$.

Thus, we get the optimal $\{p_k^{\mathrm{u}*}, \mu_k^*, \nu_k^*\}$ as shown in **Theorem** 5.1.

Bibliography

[1] P. Mell, T. Grance *et al.*, "The NIST definition of cloud computing," Computer Security Division, Information Technology Laboratory, National, 2011.

[2] A. U. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," vol. 16, no. 1, pp. 393–413, First 2014.

[3] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comp.*, vol. 13, no. 18, pp. 1587–1611, 2013.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," Tech. Rep., 2009.

[5] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, May 2010.

[6] K. Kumar and Y. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[7] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making smartphones last longer with code offload," in *Proc. ACM 8th Int. Conf. Mobile Syst. Serives (MobiSys),* San Francisco, CA, USA, Jun. 2010, pp. 49–62.

[8] S. Kosta, A. Aucinas, Pan Hui, R. Mortier, and Xinwen Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM),* Orlando, FL, USA, Mar. 2012, pp. 945–953.

[9] D. Evans, "The Internet of things: How the next evolution of the Internet is changing everything," *CISCO, White Paper*, vol. 1, no. 2011, pp. 1–11, 2011.

[10] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[11] G. P. Fettweis, "The tactile Internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[12] "Smart wireless devices and the Internet of me," *White Paper,* Juniper, Sunnyvale, CA, USA, Mar. 2015.

[13] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

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

[15] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[16] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[17] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal *et al.*, "Mobile-edge computing introductory technical white paper," *ETSI, White Paper*, Sophia Antipolis, France, 2014.

[18] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing — a key technology towards 5G," *ETSI, White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[19] G. I. Klas, "Fog computing and mobile edge cloud gain momentum open fog consortium, ETSI MEC and cloudlets," Nov. 2015.

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

[21] X. Xiang, C. Lin, and X. Chen, "Energy-efficient link selection and transmission scheduling in mobile cloud computing," *IEEE Wireless Commun. Lett.*, vol. 3, no. 2, pp. 153–156, Apr. 2014.

[22] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[23] Z. Jiang and S. Mao, "Energy delay tradeoff in cloud offloading for multi-core mobile devices," *IEEE Access*, vol. 3, pp. 2306–2316, Nov. 2015.

[24] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[25] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *proc. IEEE Int. Symp. Inf. Theory (ISIT),* Barcelona, Spain, Jul. 2016, pp. 1451–1455.

[26] O. Muñz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Tech.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[27] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[28] Y. Kim, J. Kwak, and S. Chong, "Dual-side dynamic controls for cost minimization in mobile cloud computing systems," in *Proc. 13th Int. Symp. Modeling Optim. Mobile, Ad Hoc Wireless Netw. (WiOpt),* Mumbai, India, May 2015, pp. 443–450.

[29] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

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

[31] C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[32] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2685–2700, Dec. 2013.

[33] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[34] J. Zhang, X. Hu, Z. Ning, E. C. . Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[35] H. Sun, F. Zhou, and R. Q. Hu, "Joint offloading and computation energy efficiency maximization in a mobile edge computing system," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3052–3056, Mar. 2019.

[36] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[37] X. Lyu, W. Ni, H. Tian, R. P. Liu, X. Wang, G. B. Giannakis, and A. Paulraj, "Optimal schedule of mobile edge computing for internet of things using partial information," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2606–2615, Nov. 2017.

[38] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[39] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[40] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 695–708, Jan. 2019.

[41] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4188–4200, Jun. 2019.

[42] K. Huang and V. K. N. Lau, "Enabling wireless power transfer in cellular networks: Architecture, modeling and deployment," *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 902–912, Feb. 2014.

[43] S. Bi, C. K. Ho, and R. Zhang, "Wireless powered communication: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 117–125, Apr. 2015.

[44] Q. Shi, L. Liu, W. Xu, and R. Zhang, "Joint transmit beamforming and receive power splitting for MISO SWIPT systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 6, pp. 3269–3280, Jun. 2014.

[45] M. R. A. Khandaker and K. Wong, "SWIPT in MISO multicasting systems," *IEEE Commun. Lett.*, vol. 3, no. 3, pp. 277–280, Jun. 2014.

[46] D. W. K. Ng and R. Schober, "Secure and green SWIPT in distributed antenna networks with limited backhaul capacity," *IEEE Trans. Wireless Commun.*, vol. 14, no. 9, pp. 5082–5097, Sep. 2015.

[47] H. Ju and R. Zhang, "Throughput maximization in wireless powered communication networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 1, pp. 418–428, Jan. 2014.

[48] G. Yang, C. K. Ho, R. Zhang, and Y. L. Guan, "Throughput optimization for massive MIMO systems powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 8, pp. 1640–1650, Aug. 2015.

[49] H. Ju and R. Zhang, "User cooperation in wireless powered communication networks," in *proc. IEEE Glob. Commun. Conf. (GLOBECOM),* Austin, TX, USA, Dec. 2014, pp. 1430–1435.

[50] H. Chen, Y. Li, J. L. Rebelatto, B. F. Uchoa-Filho, and B. Vucetic, "Harvest-then-cooperate: Wireless-powered cooperative communications," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1700–1711, Apr. 2015.

[51] H. Liang, C. Zhong, H. A. Suraweera, G. Zheng, and Z. Zhang, "Optimization and analysis of wireless powered multi-antenna cooperative systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3267–3281, May 2017.

[52] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1757–1771, May 2016.

[53] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[54] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.

[55] X. Sun and N. Ansari, "Green cloudlet network: A sustainable platform for mobile cloud computing," *IEEE Trans. Cloud Comput.*, vol. 8, no. 1, pp. 180–192, Jan.-March 1 2020.

[56] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.

[57] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-connected UAV: Potential, challenges, and promising technologies," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 120–127, Feb. 2019.

[58] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on UAV communications for 5G and beyond," *Proceedings of the IEEE*, vol. 107, no. 12, pp. 2327–2375, Dec. 2019.

[59] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[60] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, Dec. 2016.

[61] M. M. Azari, F. Rosas, K. Chen, and S. Pollin, "Ultra reliable UAV communication using altitude and cooperation diversity," *IEEE Trans. Commun.*, vol. 66, no. 1, pp. 330–344, Jan. 2018.

[62] J. Xu, Y. Zeng, and R. Zhang, "UAV-enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5092–5106, Aug. 2018.

[63] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, Aug. 2018.

[64] F. Zhou, R. Q. Hu, Z. Li, and Y. Wang, "Mobile edge computing in unmanned aerial vehicle networks," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 140–146, Feb. 2020.

[65] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[66] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.

[67] F. Cheng, S. Zhang, Z. Li, Y. Chen, N. Zhao, F. R. Yu, and V. C. M. Leung, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.

[68] X. Cao, J. Xu, and R. Zhang, "Mobile edge computing for cellular-connected UAV: Computation offloading and trajectory optimization," in *proc. IEEE SPAWC*, Kalamate, Greece, Jun. 2018.

[69] A. Reznik *et al.*, "Cloud RAN and MEC: A perfect pairing," *ETSI, White Paper*, no. 23, Feb. 2018.

[70] X. Hu, K. Wong, and K. Yang, "Wireless powered cooperation-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2375–2388, Apr. 2018.

[71] X. Hu, K. Wong, K. Yang, and Z. Zheng, "UAV-assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4738–4752, Oct. 2019.

[72] X. Hu, L. Wang, K. Wong, M. Tao, Y. Zhang, and Z. Zheng, "Edge and central cloud computing: A perfect pairing for high energy efficiency and

low-latency," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1070–1083, Feb. 2019.

[73] X. Hu, K. Wong, and Y. Zhang, "Wireless-powered edge computing with cooperative UAV: Task, time scheduling and trajectory design," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2020.

[74] X. Hu, K. Wong, and K. Yang, "Power minimization for cooperative wireless powered mobile edge computing systems," in *proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, USA, May 2018, pp. 1–6.

[75] X. Hu, K. Wong, K. Yang, and Z. Zheng, "Task and bandwidth allocation for UAV-assisted mobile edge computing with trajectory design," in *proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[76] X. Hu, L. Wang, K. Wong, M. Tao, Y. Zhang, and Z. Zheng, "The synergy of edge and central cloud computing with wireless MIMO backhaul," in *proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[77] X. Hu, K. Wong, and Z. Zheng, "Wireless-powered mobile edge computing with cooperated UAV," in *proc. Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC), Cannes, France*, Jul. 2019, pp. 1–5.

[78] A. U. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," vol. 16, no. 1, pp. 393–413, First 2014.

[79] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comp.*, vol. 13, no. 18, pp. 1587–1611, 2013.

[80] J. H. Christensen, "Using restful web-services and cloud computing to create next generation mobile applications," in *Proc. of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*. ACM, 2009, pp. 627–634.

[81] L. Liu, R. Moulic, and D. Shea, "Cloud service portal for mobile device management," in *proc. IEEE 7th International Conference on E-Business Engineering*. IEEE, Nov. 2010, pp. 474–478.

[82] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1107–1117, Jun. 2013.

[83] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, "Computation offloading for service workflow in mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3317–3329, Dec. 2015.

[84] Y. Nam, S. Song, and J. Chung, "Clustered NFV service chaining optimization in mobile edge clouds," *IEEE Communications Letters*, vol. 21, no. 2, pp. 350–353, 2017.

[85] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Resource allocation for blockchain-enabled distributed network function virtualization (NFV) with mobile edge cloud (MEC)," in *IEEE INFOCOM 2019 - IEEE Conference*

on Computer Communications Workshops (INFOCOM WKSHPS), 2019, pp. 1–6.

[86] H. D. Chantre and N. L. Saldanha da Fonseca, "The location problem for the provisioning of protected slices in NFV-based MEC infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1505–1514, 2020.

[87] Y. Ma, W. Liang, J. Wu, and Z. Xu, "Throughput maximization of NFV-enabled multicasting in mobile edge cloud networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 2, pp. 393–407, 2020.

[88] S. Li, Z. Guo, G. Shou, Y. Hu, and H. Li, "QoE analysis of NFV-based mobile edge computing video application," in *2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, 2016, pp. 411–415.

[89] L. Van Ma, V. Q. Nguyen, J. Park, and J. Kim, "NFV-based mobile edge computing for lowering latency of 4K video streaming," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018, pp. 670–673.

[90] M. Wang, J. Wu, G. Li, J. Li, Q. Li, and S. Wang, "Toward mobility support for information-centric IoV in smart city using fog computing," in *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, 2017, pp. 357–361.

[91] M. Wang, J. Wu, G. Li, J. Li, and Q. Li, "Fog computing based content-aware taxonomy for caching optimization in information-centric networks," in *2017*

IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2017, pp. 474–475.

[92] Y. Tang, "Minimizing energy for caching resource allocation in information-centric networking with mobile edge computing," in *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2019, pp. 301–304.

[93] C.-Y. Chang, K. Alexandris, N. Nikaein, K. Katsalis, and T. Spyropoulos, "MEC architectural implications for LTE/LTE-A networks," in *proc. ACM Workshop Mobility Evol. Internet Archit. (MobiArch),* New York, NY, USA, Oct. 2016, pp. 13–18.

[94] A. Huang, N. Nikaein, T. Stenbock, A. Ksentini, and C. Bonnet, "Low latency MEC framework for SDN-based LTE/LTE-A networks," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[95] C. Huang, M. Chiang, D. Dao, W. Su, S. Xu, and H. Zhou, "V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture," *IEEE Access*, vol. 6, pp. 17 741–17 755, 2018.

[96] Z. Lv and W. Xiu, "Interaction of edge-cloud computing based on SDN and NFV for next generation IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5706–5712, 2020.

[97] A. Hermosilla, A. M. Zarca, J. B. Bernabe, J. Ortiz, and A. Skarmeta, "Security orchestration and enforcement in NFV/SDN-aware UAV deployments," *IEEE Access*, vol. 8, pp. 131 779–131 795, 2020.

[98] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 274–291, 2020.

[99] M. Chiang, S. Ha, C. L. I, F. Risso, and T. Zhang, "Clarifying fog computing and networking: 10 questions and answers," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 18–20, Apr. 2017.

[100] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing," in *Global Internet of Things Summit (GIoTS)*, 2017, pp. 1–6.

[101] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in *Proc. ACM Workshop on Mobile Cloud Computing and Services*, 2012, pp. 29–36.

[102] ETSI White Paper No. 20: "Developing software for multi-access edge computing", Sept. 2017.

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

[104] D. Sabella *et al.*, "Toward fully connected vehicles: Edge computing for advanced automotive communications," *5G Automot. Assoc. (5GAA), White Paper*, Dec. 2017.

[105] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing." in *Proc. USENIX Conf. Hot Topics Cloud Comput. (HotCloud),* Boston, MA, USA, Jun. 2010.

[106] S. Melendez and M. P. McGarry, "Computation offloading decisions for reducing completion time," in *proc. IEEE Annu. Consum. Commun. Netw. Conf. (CCNC),* Las Vegas, NV, USA, Jan. 2017, pp. 160–164.

[107] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 4, pp. 23–32, 2013.

[108] C. E. Shannon, "Communication theory of secrecy systems," *Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.

[109] ——, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[110] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power cmos digital design," *IEICE Trans. Electron.*, vol. 75, no. 4, pp. 371–382, 1992.

[111] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Systems*, vol. 13, no. 2-3, pp. 203–221, 1996.

[112] W. Yuan and K. Nahrstedt, "Energy-efficient cpu scheduling for multimedia applications," *ACM Trans. Computer Systems*, vol. 24, no. 3, pp. 292–331, 2006.

[113] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 34–44, Jun. 2013.

[114] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing," in *proc. IEEE Glob. Commun. Conf. Workshops (GC WKSHPS),* San Diego, CA, USA, Dec. 2015.

[115] F. Ben Jemaa, G. Pujolle, and M. Pariente, "QoS-aware VNF placement optimization in edge-central carrier cloud architecture," in *proc. IEEE Global Commun. Conf. (GLOBECOM),* Washington, DC USA, Dec. 2016.

[116] M. Chen, M. Dong, and B. Liang, "Joint offloading decision and resource allocation for mobile cloud with computing access point," in *Proc. IEEE Int. Conf. Accoust. Speech Signal Process. (ICASSP),* Shanghai, China, Mar. 2016, pp. 3516–3520.

[117] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[118] Yingbin Liang and V. V. Veeravalli, "Gaussian orthogonal relay channels: Optimal resource allocation and capacity," *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 3284–3289, Sep. 2005.

[119] Y. Huo, X. Dong, T. Lu, W. Xu, and M. Yuen, "Distributed and multilayer UAV networks for next-generation wireless communication and power trans-

fer: A feasibility study," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7103–7115, Aug. 2019.

[120] J. Ouyang, Y. Che, J. Xu, and K. Wu, "Throughput maximization for laser-powered UAV wireless communication systems," in *proc. IEEE Inter. Conf. Commun. Workshops (ICC Workshops)*, Kansas City, USA, May 2018, pp. 1–6.

[121] D. Killinger, "Free space optics for laser communication through the air," *Optics and Photonics News*, vol. 13, no. 10, pp. 36–42, 2002.

[122] T. J. Nugent and J. T. Kare, "Laser power for UAVs," *LaserMotive White Paper C Power Beaming for UAVs*, 2010.

[123] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[124] R. M. Corless, G. H. Gonnet, D. E. Hare, D. J. Jeffrey, and D. E. Knuth, "On the lambertw function," *Advances in Computational mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

[125] *LTE Unmanned Aircraft Systems-Trial Report*, Qualcomm Technol., Inc., San Diego, CA, USA, May 2017.

[126] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

[127] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter*, vol. 2004, pp. 2004–2005, 2003.

[128] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming."

[129] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.

[130] D. Pompili, A. Hajisami, and H. Viswanathan, "Dynamic provisioning and allocation in cloud radio access networks (C-RANs)," *Ad Hoc Networks*, vol. 30, pp. 128–143, 2015.

[131] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K. K. Wong, R. Schober, and L. Hanzo, "User association in 5G networks: A survey and an outlook," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1018–1044, Secondquarter 2016.

[132] T. Han, Y. Han, X. Ge, Q. Li, J. Zhang, Z. Bai, and L. Wang, "Small cell offloading through cooperative communication in software-defined heterogeneous networks," *IEEE Sensors J.*, vol. 16, no. 20, pp. 7381–7392, Oct. 2016.

[133] D. Valocchi, D. Tuncer, M. Charalambides, M. Femminella, G. Reali, and G. Pavlou, "Sigma: Signaling framework for decentralized network management applications," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 616–630, Sep. 2017.

[134] G. H. Golub and C. F. Van Loan, *Matrix Computations,* 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.

[135] L. Dong, Z. Han, A. P. Petropulu, and H. V. Poor, "Improving wireless physical layer security via cooperating relays," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1875–1888, Mar. 2010.

[136] Y. Yang and M. Pesavento, "A unified successive pseudoconvex approximation framework," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3313–3328, Jul. 2017.

[137] A. Zappone and E. Jorswieck, *Energy Efficiency in Wireless Networks via Fractional Programming Theory*. Now Foundations and Trends, 2015, vol. 11, no. 3-4.

[138] E. Bjornson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: Ten myths and one critical question," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 114–123, Feb. 2016.

[139] H. Q. Ngo and E. G. Larsson, "No downlink pilots are needed in TDD massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2921–2935, May 2017.

[140] P. Harris, S. Malkowsky, J. Vieira, E. Bengtsson, F. Tufvesson, W. B. Hasan, L. Liu, M. Beach, S. Armour, and O. Edfors, "Performance characterization of a real-time massive MIMO system with LOS mobile channels," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1244–1253, Apr. 2017.

[141] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[142] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.

[143] A. He, L. Wang, Y. Chen, K. K. Wong, and M. Elkashlan, "Spectral and energy efficiency of uplink D2D underlaid massive MIMO cellular networks," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3780–3793, Sep. 2017.

[144] 3GPP TR 36.814 v9.2.0, "3rd generation partnership project: Technical specification group radio access network: Evolved univerisal terrestrial radio access (E-UTRA): further advancements for E-UTRA physical layer aspects," Mar. 2017.

[145] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[146] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[147] A. Liu and V. K. N. Lau, "Exploiting base station caching in MIMO cellular networks: Opportunistic cooperation for video streaming," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 57–69, Jan. 2015.

[148] V. Pacifici and G. Dán, "Distributed caching algorithms for interconnected operator CDNs," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 2, pp. 380–391, Feb. 2017.

[149] L. Li, G. Zhao, and R. S. Blum, "A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1710–1732, Thirdquarter 2018.

[150] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, 2014.

[151] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 22–28, 2016.

[152] E. Zeydan, E. Bastug, M. Bennis, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data caching for networking: Moving from cloud to edge," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 36–42, Sep. 2016.

[153] Q. Li, W. Shi, X. Ge, and Z. Niu, "Cooperative edge caching in software-defined hyper-cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2596–2605, 2017.

[154] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1791–1805, 2018.

[155] Chun Yuan, Yu Chen, and Zheng Zhang, "Evaluation of edge caching/off loading for dynamic content delivery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1411–1423, 2004.

[156] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[157] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 80–87, 2018.