IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 8, 2018, accepted November 22, 2018, date of publication February 1, 2019, date of current version February 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2883692

Mobile Edge Computing With Wireless Backhaul: Joint Task Offloading and Resource Allocation

QUOC-VIET PHAM[1], (Member, IEEE), LONG BAO LE[2], (Senior Member, IEEE), SANG-HWA CHUNG[3], AND WON-JOO HWANG[4], (Senior Member, IEEE)

[1]ICT Convergence Center, Changwon National University, Changwon 51140, South Korea
[2]Institut National de la Recherche Scientifique, University of Quebec, Montreal, QC H5A 1K6, Canada
[3]Department of Electrical and Computer Engineering, Pusan National University, Busan 46241 South Korea
[4]Department of Electronic, Telecommunications, Mechanical Automotive Engineering, Inje University, Gimhae 50834, South Korea

Corresponding author: Won-Joo Hwang (ichwang@inje.ac.kr)

This work was supported by the MSIT (Ministry of Science and ICT), South Korea, through the Grand Information Technology Research Center Support Program, Supervised by the Institute for Information and Communications Technology Promotion, under Grant IITP-2018-2016-0-00318.

ABSTRACT Considered as a key technology in 5G networks, mobile edge computing (MEC) can support intensive computation for energy-constrained and computation-limited mobile users (MUs) through offloading various computation and service functions to the edge of mobile networks. In addition to MEC, wireless heterogeneous networks will play an important role in providing high transmission capacity for MUs in 5G, where wireless backhaul is a cost-effective and viable solution to solve the expensive backhaul deployment issue. In this paper, we consider a setting, where MUs can offload their computations to the MEC server through a small cell base station (SBS), the SBS connects to the macro BS through a wireless backhaul, and computation resource at the MEC server is shared among offloading MUs. First, we formulate a joint optimization problem with the goal of minimizing the system-wide computation overhead. This is a mixed-integer problem and hard to derive the optimal solution. To solve this problem, we propose to decompose it into two subproblems, namely the offloading decision subproblem and the joint backhaul bandwidth and computation resource allocation subproblem. An algorithm, namely JOBCA, is proposed to obtain a feasible solution to the original problem by solving two subproblems iteratively. Finally, numerical results are conducted to verify the performance improvement of the proposed algorithm over two baseline algorithms and the close performance of the proposed algorithm compared with the centralized exhaustive search.

INDEX TERMS Computation offloading, heterogeneous networks, mobile edge computing, resource allocation, wireless backhaul.

I. INTRODUCTION

To accommodate the ever-increasing mobile traffic volume and offload the overloaded traffic from MBSs, a large number of low-cost and low-power small cell base stations (SBSs) have been deployed [1]. In such heterogeneous networks (HetNets), an important question is how to forward and receive massive traffic from SBSs to MBSs and over the core network, respectively [2]. Moreover, backhaul deployment for small cells can be based on the wired and *wireless backhauling* solutions [3], [4]. Choosing these backhauling solutions would depend on different factors, such as, the cost of implementing backhaul connections, traffic load intensity, latency, and service requirements of MUs. As small cells are densely deployed in 5G networks, deploying all small cells with wired backhaul would not be a feasible and cost-effective solution due to fiber backhaul link installation obstacles. This motivates the adoption of wireless backhaul for small cells so as to enable small cells receiving and sending data traffic to MBSs in a wireless fashion [4].

With the proliferation and popularity of mobile devices, such as, smart phones, tablets, virtual reality glass, new computation-intensive and energy-hungry applications are constantly emerging (e.g., real-time online gaming, virtual reality, natural language processing, and ultra-high-definition video streaming). However, since mobile devices are often equipped with low-capacity battery and limited-computation capability, they may not run many of these applications efficiently and become a bottleneck for the future development of mobile applications. One of the possible solutions is enabling mobile devices to *offload* their intensive computation tasks to the remote cloud center, which has high computational capability and large storage capacity [5]. Nevertheless, existing mobile cloud computing faces various challenges including high latency due to the long propagation distance from mobile devices to the remote cloud center, low scalability, and high burden on fronthaul links due to the centralized deployment of the cloud center. To address the drawbacks of mobile cloud computing, mobile edge computing (MEC) has been proposed and developed by the European Telecommunications Standards Institute (ETSI) to "offer application developers and content providers cloud-computing capabilities and an IT service environment at the edge of the network" [6]. The key idea behind the MEC concept is to move the cloud services, resources, and functions to the network edges. As opposed to mobile cloud computing, MEC is able to achieve lower latency and higher reliability and energy efficiency [7], [8], which is therefore suitable for ultra-reliable and low-latency applications in the emerging 5G networks.

A similar concept to the MEC is the fog computing, which has been introduced by Cisco in 2012 as a supplement to mobile cloud computing. MEC and Fog have some common characteristics. First, both MEC and Fog are able to provide low latency and location awareness to the end users at the network edges. In addition, MEC and Fog are usually distributed over widespread geographic areas instead of the centralized implementation as in the cloud radio access network and mobile cloud computing [8]. Finally, due to close proximity to the end users, both Fog and MEC are suitable for latency-sensitive applications. However, there are two significant differences between MEC and Fog. The first difference is that the fog computing was developed by Cisco in 2012 while MEC was introduced by ETSI in 2014. Another important difference is that fog nodes are not integrated to the mobile networks and the fog computing is highly favoured by the service providers. However, MEC servers are deployed by the network operators as a part of the mobile networks and the MEC is highly favoured by the telecoms service providers and/or the telecoms infrastructure providers, which would have their own backbone and radio networks. Since the bandwidth resource allocation of wireless backhaul is taken into consideration, the proposed design in our current work is more suitable for MEC networks.

Due to the great potentials of HetNets and MEC, many research studies have been conducted for wireless backhaul [2]–[4], [9]–[13] and mobile edge computing [7], [8], [14]–[22]. Among major issues in mobile edge computing, computation offloading is of central importance. However, computation offloading may incur additional overhead in terms of energy consumption and latency, e.g., the local execution only suffers from the locally processing delay whereas the remote execution latency includes the transmission delay of the incurred data from mobile users to the MEC server, the remotely processing delay at the MEC server, and the response delay required by the MEC server to send back the result to the users [20], [22]. In the presence of multiple users, the MEC server must be able to simultaneously execute multiple computation tasks and the scarce wireless bandwidth needs to be shared among multiple users. Compared to the resourceful cloud, the MEC server usually has finite resources and would not be able to meet all users' computation requirements. As a consequence, the joint optimization of offloading decisions and resource allocation is an important research problem in MEC systems to improve the network performance. Even though there are numerous studies on computation offloading for MEC systems, aforementioned works a) generally assume that small cells are connected with macro cells by wired backhaul such as fiber and optical links, b) have not studied the problem of joint backhaul and access bandwidth allocation, and c) do not take the offloading time over wireless backhaul link into consideration. In fact, existing studies on MEC networks have not imposed any resource constraints on the wireless backhaul links. In practice, such wireless backhaul constraints exist in wireless MEC HetNets and the backhaul link capacities strongly impact the offloading decisions. While the wireless backhaul bandwidth allocation problem has been studied before to optimize energy efficiency [12] and spectral efficiency [4], [10] in HetNets, we are not aware of any work that addresses the optimization of *computation offloading* and *resource (communication and computation) allocation* in MEC systems with *wireless backhaul* consideration.

The main contribution of this paper is to introduce a novel framework for joint computation offloading and resource allocation in MEC networks with wireless backhaul. The important factors, such as offloading decisions, computation resource, and bandwidth spectrum allocation, are jointly considered. Here, computation offloading pertains to finding the offloading decisions for users and resource allocation relates to the computation resource allocation at the MEC server and bandwidth resource sharing between the wireless access transmission and wireless backhaul transmission. To the best of our knowledge, this work is the first attempt to investigate the joint *computation offloading decisions* and *resource allocation* problem with *wireless backhaul* in the mobile edge computing system. In a nutshell, the key features and contributions of our proposed design can be summarized as follows:

- We formulate a joint optimization problem of offloading decision, wireless backhaul bandwidth partitioning, and computation resource allocation, which has not been studied before. An MEC system with an MBS, an SBS, and multiple MUs is studied, where wireless backhaul is used to establish the connection between the SBS and MBS (MEC server), available spectrum is shared between the access links, i.e., between MUs and SBS, and the wireless backhaul transmission, i.e., between SBS and MBS/MEC server. The formulated optimization problem aims to minimize the system-wide computation overhead subject to constraints on offloading decision, computation resource at the MEC server, and bandwidth allocation. This is a mixed integer and NP-Hard optimization problem, which is hard to obtain the global centralized optimal solution.

- To solve the underlying optimization problem, we devise a suboptimal algorithm by decomposing the original problem into two subproblems: the first one optimizes the computation offloading decision while the other optimizes the backhaul bandwidth partitioning and computation resource allocation, which is further decomposed into subproblems of backhaul bandwidth allocation and computation resource allocation at the MEC server. Then, we solve these subproblems individually and propose an iterative algorithm, namely JOBCA, to achieve the solution to the original problem.
- Our simulation results confirm that our proposed algorithm can achieve better performance in comparison with two baseline schemes in terms of the percentage of offloading users and system-wide computation overhead, and has similar performance with that of the exhaustive search (i.e., there is a small optimality gap).

The rest of our paper is organized as follows. In Section II, we briefly introduce the background in mobile edge computing and wireless backhaul in 5G networks, and summarize related studies on computation offloading and wireless backhaul. In Section III, we consider a network model and formulate the considered optimization problem. The proposed algorithm is described in Section IV. Section V discusses our proposed algorithm in ultra-dense networks (UDNs) with inter-cell interference and with partial computation offloading, and briefly considers deep reinforcement learning (DRL) for computation offloading in a dynamic MEC system with wireless backhaul. We provide simulation results in Section VI while concluding the paper and providing some interesting future directions in Section VII.

II. BACKGROUND AND RELATED WORK

A. MOBILE EDGE COMPUTING

The development of mobile edge computing is based on different related technologies including mobile cloud computing [5], private cloud [23], cloudlet [24], and fog computing [25]. The key objective design of mobile edge computing is to distribute cloud contents, services, and resources to mobile devices in a closer proximity. According to the ETSI white paper [26], mobile edge computing can be characterized by some features, namely on-premises, proximity, lower latency, location awareness, and network context information. These features can be shortly explained, as follows:

- **On-premises**: edge computing can operate independently from the rest of the network and has access to local resources.
- **Proximity**: the edge is located closely to mobile devices and it is able to access mobile devices directly.
- **Lower latency**: thanks to the short distance to mobile devices, edge computing can achieve relatively low latency, which can efficiently support emerging latency-critical applications such as autonomous driving, virtual sports, and real-time online gaming.
- **Location awareness**: due to close proximity, the MEC system can determine locations of mobile devices by requesting low-level signaling information.
- **Network context information**: being able to access to local information and close to mobile devices, the MEC system can run applications and services with real-time network information, e.g., cell load and subscriber location.

Over the last few years, many research studies have been conducted to realize enormous potentials of mobile edge computing in different network scenarios. Three examples are provided in the following as demonstration. Firstly, in order to meet critical requirements of ultra-reliable and low-latency applications in 5G networks, authors in [27] and [28] considered the joint problem of the latency and reliability based computation offloading optimization in MEC systems. Secondly, it is forecast that there will be billions of Internet of Things (IoT) devices in 5G networks, where each IoT device is limited in storage and computation resources. By offloading computations to the MEC servers, IoT devices can prolong their battery life and reduce their energy consumption [7]. Thirdly, it can be expected that edge servers will be densely deployed in 5G, where each server is equipped with an energy-limited battery. Therefore, energy harvesting based MEC systems are promising, where edge servers and mobile devices are powered by harvesting energy from external sources, e.g., solar radiation and wind energy [29].

By offloading computations to the MEC servers, MUs are able to exploit the rich computation resource from the edge servers and relieve their limitations on storage, computing, and computation. Over the past few years, a number of studies have been carried out to address the computation offloading, e.g., [7], [8], [14]–[19]. According to recent surveys [7], [8], there are generally two mains types of computation offloading: binary offloading and partial offloading. In the binary offloading, as considered in our paper, a computation task cannot be partitioned into sub-tasks and the whole task must be executed either locally at the MU or remotely at the MEC server. Whereas, in partial offloading, a task can be divided into sub-tasks, which can be executed at different MEC servers [7]. Chen *et al.* [14] showed that the problem of finding the maximum number of offloading users is NP-Hard and the authors adopted a game-theoretic approach to find the optimal offloading decision in a distributed manner. The authors in [15] and [16] considered the dynamic voltage scaling technique for computation offloading under different design objectives and scenarios, e.g., the offloading ratio for local and remote computing in a single-task MEC system [15] and the offloading decision for one MU with multiple tasks [16].

A joint computation offloading and interference management framework in HetNets was proposed in [17]. Lyu *et al.* [18] and Pham *et al.* [20] studied problems of transmit power and offloading decision for MUs and computation resource at the single MEC server and multiple MEC servers, respectively. Using the submodular optimization, a heuristic semi-distributed algorithm was proposed in [18], and matching theory was utilized to devise a decentralized computation offloading scheme in [20]. Motivated by the fact that existing computation offloading frameworks only take resources of MUs and MEC servers into consideration, Guo and Liu [19] proposed an architecture that supports the coexistence of the centralized cloud computing center and distributed mobile edge computing servers. In addition, the authors proposed three collaborative offloading solutions, where each computation task can be either executed locally or offloaded to the edge and centralized cloud center for execution.

B. WIRELESS BACKHAUL

To fulfill key requirements of the 5G networks (e.g., 1000 times higher data rate, sub-millisecond latency, and 10 times higher energy efficiency), dense deployment of small cells will be one of the key solutions. According to [2] and [3], one fundamental question is how to design efficient backhaul solutions with capability to forward and receive massive traffic from/to small cell users. With the increasing network densification, implementing wired backhaul for a great number of small cells would not be affordable or even feasible. The reasons for this are as follows:

- Although wired backhaul approaches can provide higher reliability and data rate compared with wireless backhaul solutions, it is costly to implement wired backhaul for all small cells and time-consuming to deploy connections for a large number of small cells [2], [3]. In addition, deployment of wired backhaul depends on various factors such as the location of small cells, quality of service (QoS) requirements of MUs.
- Different frequency bands have been proposed for wireless backhaul [3], [30], such as cellular frequency band, millimeter wave (mmWave) band, sub-6 GHz band, satellite frequency band, and TV white space band. Therefore, wireless backhauling provides a practical solution for dense small cells in the emerging 5G networks.
- Providing wireless access to rural/remote areas and some urban areas requires to carefully consider the deployment cost [31]. In such scenarios, wireless backhaul is a practical and affordable solution, which can simplify the deployment and drive down the maintenance cost.
- Besides the improvements in data rate, reliability, and latency of communication, the emerging 5G networks will better support emergency services, which have the stringent requirements on response time [32]. Nevertheless, in HetNets with all wired backhaul links, broken wired backhaul may not be recovered instantly; therefore, emergency services can be severely impacted due to the slow network recovery and low reliability [33]. Deployment of small cells with wireless backhaul can enable to mitigate the aforementioned issue.

Recently, research on various issues related to wireless backhaul for HetNets has received enormous attention from the communication community. In [4], a joint optimization problem of transmit beamforming, power allocation, and bandwidth partitioning in HetNets was considered. The authors studied the reverse time division duplexing (RTDD) system, which is to calibrate the transmission of small cells and a macro cell in two consecutive time slots, and proposed to partition the bandwidth for two consecutive time slots using two separate partitioning factors. The user association problem for wireless networks with wireless backhaul was studied in [10] and [13]. Liu *et al.* [10] considered a massive multiple-input multiple-output HetNet, where each MU can associate with either a pico cell or the macro cell, and this work jointly optimized the association vector and bandwidth allocation factor. A joint resource allocation and user association problem with backhaul constraints was considered for hybrid-energy-powered HetNets, where the base station can be powered by the traditional grid, renewable energy sources, or both. The energy efficiency of HetNets with wireless backhaul was considered in [9], [11], and [12]. Utilizing the RTDD as in [4], the authors in [9] supposed that the bandwidth is equally partitioned for wireless access/backhaul communication and the work optimized a new metric, called access energy efficiency, in which due to the consideration of wireless backhaul, the adaptive decoding power at SBSs are taken into consideration. A holistic approach to energy efficiency optimization was studied in [11], where both the access and backhaul transmissions are considered. By jointly optimizing the transmit power of SBSs and unified bandwidth partitioning factor, Zhang *et al.* [12] investigated the energy efficiency maximization problem under constraints of backhaul capacity and user QoS requirements.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. NETWORK MODEL

We consider a network setting as illustrated in Fig. 1, where a macro cell with one MBS is overlaid by one small cell (with one SBS), and N MUs are randomly positioned in and associated with the SBS. We assume that the small cell shares the same spectrum with the macro cell and the MBS is collocated with an MEC server to provide computation services to MUs. In this work, the wireless access transmission refers to the transmission between MUs and the SBS, whereas the wireless backhaul transmission refers to the communication between the SBS and MBS. In addition, we assume that α is the fraction of total available bandwidth allocated for the wireless access transmission, i.e., $(1-\alpha)$ fraction of bandwidth is allocated for the wireless backhaul transmission. To enable tractable analysis, we assume that computation offloading through the SBS can be realized at the beginning of each offloading period [34]. Latency-tolerant applications (e.g., natural language processing and face recognition) are considered in our work, where the offloading period is within several seconds [35]. This assumption could ensure that the offloading period is at the shorter timescale than that of the

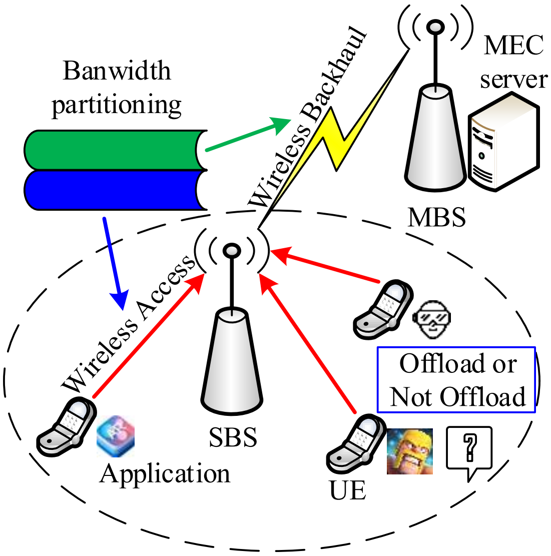


FIGURE 1. The model for multiple MUs, one SAP, and one MBS.

network dynamics and UE mobility, and the computation task can be accomplished within the offloading period [34], [36].

B. COMMUNICATION MODEL

We consider a quasi-static network scenario, where MUs' locations remain unchanged during the computation offloading period but they can change locations over different periods. We assume that different MUs use different subchannels so that intra-cell interference is eliminated. Moreover, the inter-cell interference is ignored due to orthogonal bandwidth allocation over the cells and/or wall penetration loss and low transmit power of SBSs, especially when small cells are utilized in sparse wireless networks and the interference received from adjacent small cells is negligible.

Denote by p_n and h_n as the fixed transmit power of MU n and the channel gain between MU n and the SBS, respectively. The signal-to-noise ratio of MU n is expressed by $\gamma_n = p_n h_n / n_0$, where n_0 is the additive white Gaussian noise (AWGN) power. Accordingly, the transmission rate between MU n and the SBS is given by $r_n = \frac{\alpha}{N_{\text{off}}} W \log_2 (1 + \gamma_n)$, where N_{off} is the number of offloading MUs and W is the system bandwidth (i.e., we assume equal bandwidth allocation for different MUs in each cell). Similarly, the transmission rate of the wireless backhaul from the SBS to MBS can be written as $R_{\text{bh}} = (1-\alpha)\, W \log_2 (1 + P_0 h_0 / n_0)$, where P_0 is the fixed transmit power of the SBS and h_0 is the channel gain between the SBS and MBS.

C. COMPUTATION MODEL

Each MU n has a computation task $I_n = \{D_n, C_n\}$, where D_n is the computation input data size (in bits) and C_n is the number of CPU cycles required to accomplish the task, i.e., computation workload/density. Each computation task can be executed either locally or remotely in the MEC server.

For local computing resource, we denote f_n^l as the computational capability (in CPU cycles per second) of MU n, where the superscript l stands for *local*. We consider a heterogeneous computing scenario where MUs may have different computational capabilities. Let t_n^l be the completion time of the task I_n by MU n, which can be computed as $t_n^l = \frac{C_n}{f_n^l}$. To compute the energy consumption E_n^l (in Joule) of MU n when the task is executed locally, we adopt the model in [7], [17], and [18]. Specifically, $E_n^l = \kappa_n C_n \left(f_n^l\right)^2$, where κ_n is a coefficient relating to the chip's hardware architecture. According to the measurements in [17], we set $k_n = 5 \times 10^{-27}$. It is worth noting that t_n^l and E_n^l depend on unique features of MU n and the underlying application; therefore, they can be computed in advance.

The computation overhead[1] by the local computing approach which is a function of the computational time and energy consumption is defined as [14], [17], [20]

$$Z_n^l = \lambda_n^t t_n^l + \lambda_n^e E_n^l, \tag{1}$$

where $\lambda_n^t \in [0, 1]$ and $\lambda_n^e \in [0, 1]$ are respectively weighted parameters[2] for the computational time and energy consumption of MU n. In this paper, we employ the weighted sum method to deal with the multi-objective optimization problem of computational time and energy as in (1). In general, the objective function can be defined using other approaches, for example, lexicographic method, weighted max-min method, and weighted product method [1], [38] which is outside of the scope of this paper and will be studied in future work. Similar to the heterogeneous computation tasks of MUs, different MUs may have different values of λ_n^t and λ_n^e. The weighted parameters can affect the offloading decisions of MUs. Consider a network scenario with three MUs as an example, the first MU with a latency-sensitive application sets $\lambda_n^t = 1$ and $\lambda_n^e = 0$, the second MU running an energy-hungry application and low battery state can set the weighted parameters as $\lambda_n^t = 0$ and $\lambda_n^e = 1$, the third MU can set $0 < \lambda_n^t, \lambda_n^e < 1$ if it takes both computational time and energy consumption in making offloading decision. It is worth noting that an MU may have different weighted parameters λ_n^t and λ_n^e for different applications and the weighted parameters can be dynamically changed for different computation offloading periods due to the dynamic computation demands of MUs.

If an MU is not able to execute the computation task due to the limited battery or stringent application requirements, it will offload the computation task to the MEC server. By offloading, an MU incurs the extra overhead in terms of the time and energy consumption. The overhead in time comprises the transmission time from MU/SBS to

[1] Here, the term "overhead" means the execution cost of the computation task. Since binary offloading is considered in this paper, local computation overhead refers to the case of local execution and remote computation overhead refers to the case of computation offloading (remote execution). The term "cost" can be used in the same manner.

[2] The proper values of weighted parameters λ_n^t and λ_n^e can be determined using the multiple criteria decision making theory [14], [37]. To normalize λ_n^t and λ_n^e, some approaches can be utilized such as dividing λ_n^t by the local completion time and λ_n^e by the local energy consumption [18], and defining units of λ_n^t and λ_n^e as monetary unit per second and joule, respectively.

SBS/MBS and the execution time at the MEC server, whereas the overhead in energy consumption includes the energy for computation offloading from MU/SBS to SBS/MBS. Here, we ignore latency in downlink transmission of the computational result due to the small size of the involved data and we also ignore the energy consumption at the MEC server since it is generally powered by cable power supply [14], [17].

The time and energy costs for computation offloading from the MU n to the SBS are, respectively, computed as

$$t_n^{\mathrm{ac}} = \frac{D_n}{\alpha N_{\mathrm{off}}^{-1} W \log_2 \left(1 + \frac{p_n h_n}{n_0}\right)}, \tag{2}$$

where the superscript "ac" stands for *access* and

$$E_n^{\mathrm{ac}} = p_n t_n^{\mathrm{ac}} = \frac{p_n D_n}{\alpha N_{\mathrm{off}}^{-1} W \log_2 \left(1 + \frac{p_n h_n}{n_0}\right)}. \tag{3}$$

Similarly, the time and energy costs for computation offloading from the SBS to the MBS are given, as follows:

$$t_n^{\mathrm{bh}} = \frac{D_n}{(1-\alpha)\, W \log_2 \left(1 + \frac{P_0 h_0}{n_0}\right)}, \tag{4}$$

where the superscript "bh" stands for *backhaul* and

$$E_n^{\mathrm{bh}} = P_0 t_n^{\mathrm{bh}} = \frac{P_0 D_n}{(1-\alpha)\, W \log_2 \left(1 + \frac{P_0 h_0}{n_0}\right)}. \tag{5}$$

The MEC server provides each offloading MU a computing f_n^r (in cycles per second). Then, the execution time of the computation task I_n at the MEC server is expressed as $t_n^{\mathrm{exe}} = C_n / f_n^r$.

Similar to the computation overhead due to local execution, the computation overhead under remote execution can be computed as $Z_n^r = \lambda_n^t \left(t_n^{\mathrm{exe}} + t_n^{\mathrm{ac}} + t_n^{\mathrm{bh}}\right) + \lambda_n^e \left(E_n^{\mathrm{ac}} + E_n^{\mathrm{bh}}\right)$.

D. PROBLEM FORMULATION

We define the offloading decision profile as $\boldsymbol{x} = \{x_n, \forall n \in \mathcal{N}\}$ and computation resource vector as $\boldsymbol{f} = \{f_n^r, \forall n \in \mathcal{N}\}$. Since our design aims to minimize the system-wide computation overhead, the objective function is defined as $Z(\boldsymbol{x}, \alpha, \boldsymbol{f}) = \sum_{n \in \mathcal{N}} Z_n(x_n, \alpha, \boldsymbol{f})$, where $Z_n(x_n, \alpha, f_n^r) = (1 - x_n)\, Z_n^l + x_n Z_n^r, \forall n \in \mathcal{N}$. The joint problem of offloading decision, wireless backhaul bandwidth partitioning, and computation resource allocation is formulated as follows:

$$\begin{aligned} \min_{\{\boldsymbol{x}, \alpha, \boldsymbol{f}\}} \ & \sum\nolimits_{n \in \mathcal{N}} Z_n(x_n, \alpha, f_n^r) \\ \text{s.t. } & \mathrm{C1}: x_n = \{0, 1\}, \quad \forall n \in \mathcal{N} \\ & \mathrm{C2}: 0 \le \alpha \le 1, \\ & \mathrm{C3}: \sum\nolimits_{n \in \mathcal{N}_{\mathrm{off}}} r_n \le R_{\mathrm{bh}}, \\ & \mathrm{C4}: f_n^r > 0, \quad \forall n \in \mathcal{N}_{\mathrm{off}} \\ & \mathrm{C5}: \sum\nolimits_{n \in \mathcal{N}_{\mathrm{off}}} f_n^r \le f_0. \end{aligned} \tag{6}$$

In this formulation, C1 represents the binary offloading decisions of computation tasks and C2 captures the lower and upper bounds of the bandwidth partitioning factor α; $\alpha = \{0, 1\}$ when there is no offloading MUs. C3 enforces that the wireless backhaul transmission rate from the SBS to the MBS should be greater than the total data access transmission rate on the uplink between the SBS and its associated MUs. Since the SBS receives computation tasks from MUs and transmits the received tasks to the MEC server, the wireless backhaul link becomes a crucial factor for the offloading rates from MUs, thus affecting the offloading decisions. Even if MUs decide to offload and have good channel qualities, offloading all the tasks would not be preferable when the wireless backhaul link is of low capacity. Here, $N_{\mathrm{off}} = \sum_{n \in \mathcal{N}} x_n$ and $p_n = 0$ if the MU n processes its computation task locally. The last two constraints imply that the required computation resource of offloading MUs is positive and the total computation resource assigned to offloading MUs is limited by the maximum computational capability f_0 of the MEC server, respectively.

Remark 1: Since the offloading decision variables $\boldsymbol{x}$ are binary and the bandwidth partitioning factor α and computation resource $\boldsymbol{f}$ are continuous, the optimization problem (6) is a mixed integer problem (MIP). According to [39], the MIP is NP-hard in general complexity theory. Some general algorithms can be used for solving MIP, for example, branch-and-bound algorithm, branch-and-cut method, and exhaustive search. However, these algorithms often have prohibitive time complexity and would only be feasible for the network scenarios with a small number of MUs. To tackle the formulated problem under more general settings, e.g., the Internet of Things with a massive number of IoT devices, we propose to decompose the original problem into subproblems and then solve them separately and iteratively until convergence. Many existing works have applied the decomposition technique for solving their problems with convergence and good performance, for example, [10]–[12], [40].

Remark 2: For wireless networks using wired backhaul, there is no need to partition the bandwidth for wireless access and wireless backhaul transmissions and the entire bandwidth can be allocated for the wireless access transmission, i.e., $\alpha = 1$. The performance of MEC systems with wired backhaul is higher than that with wireless backhaul since there is no cost of time and energy for computation offloading over the wireless backhaul, thus lowering the remote computation overhead of the MUs and increasing the percentage of offloading users. In that case, the constraints C2 and C3 are absent from the optimization problem (6). Moreover, the resource allocation subproblem is to assign computing resource of the MEC server to the offloading users and one does not need to check the backhaul capacity constraint when making the offloading decision (since the capacity of wired backhaul links is usually large enough and sometimes assumed to be unlimited).

IV. PROPOSED ALGORITHM

We decompose the original problem into two subproblems: offloading decision for a given bandwidth factor and computation resource allocation, and joint wireless backhaul bandwidth and computation resource allocation for a fixed offloading decision. Then, two subproblems are solved individually and an iterative algorithm is proposed to solve the original problem. The proposed framework for solving the original optimization problem (6) is summarized in Fig. 2.

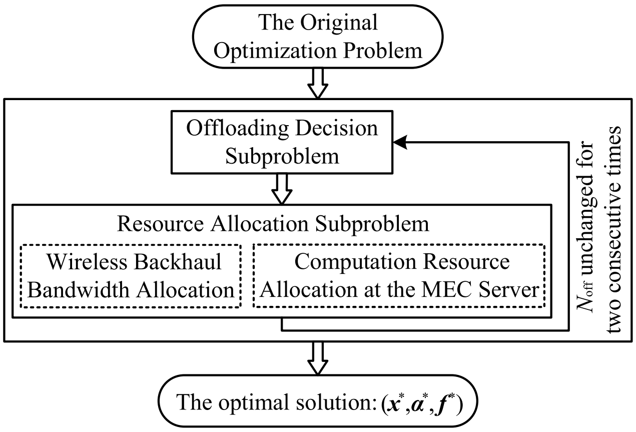

The Original Optimization Problem
Offloading Decision Subproblem
Resource Allocation Subproblem
Wireless Backhaul Bandwidth Allocation
Computation Resource Allocation at the MEC Server
$\mathcal{N}_{\rm off}$ unchanged for two consecutive times
The optimal solution: $(\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{f}^*)$

FIGURE 2. **Proposed framework for solving the problem (6).**

A. OFFLOADING DECISION

When the bandwidth allocation factor α and computation resource $\boldsymbol{f}$ are both fixed, the offloading decision subproblem can be rewritten as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}} \sum\nolimits_{n\in\mathcal{N}} Z_n(x_n) \\ &\text{s.t. } x_n = \{0, 1\}, \quad \forall n \in \mathcal{N} \\ &\qquad \sum\nolimits_{n\in\mathcal{N}_{\rm off}} r_n(\boldsymbol{x}) \le R_{\rm bh}. \end{aligned} \tag{7}$$

The objective function of (7) can be re-expressed as $\sum_{n\in\mathcal{N}} Z_n(x_n) = \sum_{n\in\mathcal{N}} x_n \left(Z_n^r - Z_n^l\right) + \sum_{n\in\mathcal{N}} Z_n^l$, where the second part is known in advance; therefore, this part can be eliminated from the optimization problem (7). Recall that this paper considers binary offloading such that a computation task cannot be partitioned into subtasks. To be executed remotely, computation offloading has to be profitable to MUs in terms of computation overhead. Consequently, MU n decides to offload its computation task I_n to the MEC server only if $\left(Z_n^r - Z_n^l\right) \le 0$, i.e., $Z_n^r \le Z_n^l$ or the local computation overhead of the MU n is greater than that of the MU n when the computation task I_n is executed remotely at the MEC server. In addition, for a given bandwidth allocation factor α, $R_{\rm bh}$ is fixed, whereas the total access rate is dependent on the offloading decisions $\boldsymbol{x}$. Consequently, for a given $(\alpha, \boldsymbol{f})$, the last constraint in (7) somehow relates to the maximum number of offloading MUs. The problem (7) can be viewed as a many-to-one matching game, where several MUs send their bid requests to the MEC server, which either accepts or rejects the offloading requests from MUs according to the computation allocation policy, so as to minimize the system-wide computation overhead.

Inspired by the matching theory, we propose an algorithm to solve the offloading decision problem (7), which can be described as follows. Initially, all N MUs are assumed to be in the offloading mode and send their requests to the MEC server, which is responsible for either accepting or rejecting the computation offloading request from MUs. At each iteration t, if $Z_n^r > Z_n^l$ (i.e., MU n does not benefit from computation offloading), this MU does not offload its computation task I_n to the MEC server, i.e., $x_n(t) = 0$. Only MUs that satisfy the offloading condition $Z_n^r \le Z_n^l$ can offload their computation tasks to the MEC server, i.e., $x_n(t) = 1$. By checking the offloading conditions for all the MUs, we can determine the outputs, which are the offloading decision vector $\boldsymbol{x}(t)$, the set of offloading MUs $\mathcal{N}_{\rm off}(t)$, and the set of rejected MUs $\mathcal{N}_{\rm rej}(t)$, all at the iteration t. Beside ensuring that the offloading MUs benefit from the remote execution, one must satisfy the backhaul capacity constraint. When offloading all the MUs in $\mathcal{N}_{\rm off}(t)$ violates the backhaul capacity constraint, some MUs, even satisfying the offloading condition $Z_n^r \le Z_n^l$, are not accepted to offload their computation tasks to the MEC server. In this case, the MUs in $\mathcal{N}_{\rm off}(t)$ are sequentially removed from the set of offloading MUs in the ascending order of the computation gain $\left(Z_n^l - Z_n^r\right)$. The removed users, so-called $\mathcal{N}_{\rm rem}$, are then inserted into the set of rejected users, i.e., $\mathcal{N}_{\rm rej}(t) := \mathcal{N}_{\rm rej}(t) \cup \mathcal{N}_{\rm rem}$ and $\mathcal{N}_{\rm off}(t) := \mathcal{N}_{\rm off}(t) \setminus \mathcal{N}_{\rm rem}$. Then, the set of offloading MUs $\mathcal{N}_{\rm off}(t)$ will be used as the input to the optimization problem of computation resource allocation and bandwidth spectrum partitioning, as presented in the next subsection IV-B, while the rejected MUs $\mathcal{N}_{\rm rej}(t)$ will be processed in the next iteration $(t + 1)$.

B. JOINT WIRELESS BACKHAUL BANDWIDTH AND COMPUTATION RESOURCE ALLOCATION

Once the solution $\mathcal{N}_{\rm off}(t)$ is achieved for the subproblem (7) parameterized by $(\alpha, \boldsymbol{f})$, it is used for the wireless backhaul bandwidth and computation resource allocation as follows:

$$\begin{aligned} &\min_{\{\alpha, \boldsymbol{f}\}} \sum\nolimits_{n\in\mathcal{N}_{\rm off}} Z_n\left(\alpha, f_n^r\right) \\ &\text{s.t. } 0 \le \alpha \le 1, \\ &\qquad \sum\nolimits_{n\in\mathcal{N}_{\rm off}} r_n(\alpha) \le R_{\rm bh}, \\ &\qquad f_n^r > 0, \quad \forall n \in \mathcal{N}_{\rm off} \\ &\qquad \sum\nolimits_{n\in\mathcal{N}_{\rm off}} f_n^r \le f_0. \end{aligned} \tag{8}$$

The objective function can be rewritten as

$$\sum_{n\in\mathcal{N}_{\rm off}} Z_n\left(\alpha, \boldsymbol{f}\right) = \sum_{n\in\mathcal{N}_{\rm off}} \frac{\lambda_n^t C_n}{f_n^r} + \frac{A}{\alpha} + \frac{B}{1-\alpha}, \tag{9}$$

where $t_n^{'\rm ac} = t_n^{\rm ac}\alpha$, $t_n^{'\rm bh} = t_n^{\rm bh}(1-\alpha)$, $E_n^{'\rm ac} = E_n^{\rm ac}\alpha$, and $E_n^{'\rm bh} = E_n^{\rm bh}(1-\alpha)$, $A = \sum_{n\in\mathcal{N}_{\rm off}} \left(\lambda_n^t t_n^{'\rm ac} + \lambda_n^e E_n^{'\rm ac}\right)$, and $B = \sum_{n\in\mathcal{N}_{\rm off}} \left(\lambda_n^t t_n^{'\rm bh} + \lambda_n^e E_n^{'\rm bh}\right)$. It can be observed from constraints in (8) and the equivalent objective function (9) that the joint problem of bandwidth and computation resource allocation is decoupled in the bandwidth partitioning factor α and computation resource vector $\boldsymbol{f}$. Consequently, the problem (8) can be further decomposed into two subproblem: one for wireless bandwidth backhaul allocation and the other for computation resource at the MEC server, which will be solved in the following.

1) WIRELESS BACKHAUL BANDWIDTH ALLOCATION

In (8), the second constraint can be equivalently represented as

$$\alpha \leq \alpha_{\text{up}} := \frac{\log_2\left(1+\frac{P_0 h_0}{n_0}\right)}{\log_2\left(1+\frac{P_0 h_0}{n_0}\right)+\sum\limits_{n\in\mathcal{N}_{\text{off}}}\frac{1}{N_{\text{off}}}\log_2\left(1+\frac{p_n h_n}{n_0}\right)}.$$

Therefore, the feasible set of the optimization problem reduces to $0 \leq \alpha \leq \alpha_{\text{up}}$ since $\alpha_{\text{up}} < 1$ if there are more than one offloading MU. After decomposition of (8), the optimal bandwidth allocation factor can be obtained by solving the following problem

$$\min_{\alpha}\left[f(\alpha) = \frac{A}{\alpha} + \frac{B}{1-\alpha}\right]$$
$$\text{s.t. } 0 \leq \alpha \leq \alpha_{\text{up}}. \tag{10}$$

The second-order derivative of $f(\alpha)$ can be expressed as

$$\frac{\partial^2 f}{\partial \alpha^2} = \frac{2A}{\alpha^3} + \frac{2B}{(1-\alpha)^3}.$$

It can be verified that $\frac{\partial^2 f}{\partial \alpha^2} > 0$ with $0 < \alpha \leq \alpha_{\text{up}}$; therefore, the optimization problem (10) is convex. Hence, the optimal value of the bandwidth partitioning factor α can be achieved by letting $\frac{\partial f}{\partial \alpha} = 0$ and comparing the results with the boundary values $0 \leq \alpha \leq \alpha_{\text{up}}$.

2) COMPUTATION RESOURCE ALLOCATION AT THE MEC SERVER

From (8), the problem of computation resource allocation at the MEC server can be rewritten as

$$\min_{\boldsymbol{f}}\left[g(\boldsymbol{f}) = \sum_{n\in\mathcal{N}_{\text{off}}}\frac{\lambda_n^t C_n}{f_n^r}\right]$$
$$\text{s.t. } f_n^r > 0, \quad \forall n \in \mathcal{N}_{\text{off}}$$
$$\sum\nolimits_{n\in\mathcal{N}_{\text{off}}} f_n^r \leq f_0. \tag{11}$$

It can be verified that $\partial^2 g/\partial f_n^{r2} = 2C_n/f_n^{r3}$ and $\partial^2 g/\partial f_n^r \partial f_m^r = 0$ for all $n \neq m$; hence, the objective in (11) is a convex function. Additionally, two constraints in (11) are both linear. Therefore, (11) is a convex problem and the optimal solution can be easily obtained by the duality technique [38], [41].

Let ν be the dual vector associated with the second constraint, the Lagrangian can be written as

$$L(\boldsymbol{f},\nu) = \sum_{n\in\mathcal{N}_{\text{off}}}\frac{\lambda_n^t C_n}{f_n^r} + \nu\left(\sum_{n\in\mathcal{N}_{\text{off}}} f_n^r - f_0\right).$$

Then, the dual function is defined as $G(\nu) = \min\limits_{\boldsymbol{f}\succ\boldsymbol{0}} L(\boldsymbol{f},\nu)$ and the dual problem is given by $\max\limits_{\nu>0} G(\nu)$. Since (11) is a convex problem, the optimal computation resource f_n^r can be derived by setting the first-order derivative of $L(\boldsymbol{f},\nu)$ with respect to (w.r.t.) f_n^r to zero. Accordingly, we have $f_n^r = \sqrt{\lambda_n^t C_n/\nu}$, from which we have $G(\nu) = 2\sum_{n\in\mathcal{N}_{\text{off}}}\sqrt{\lambda_n^t C_n \nu} - \nu f_0$. By setting the first-order derivative of $G(\nu)$ w.r.t. ν to zero and plugging back to the formula of f_n^r, the optimal computation resource can be written as follows:

$$f_n^{r*} = \frac{f_0\sqrt{\lambda_n^t C_n}}{\sum\limits_{n\in\mathcal{N}_{\text{off}}}\sqrt{\lambda_n^t C_n}}. \tag{12}$$

From (12), the computation resource assigned to the MU n is proportional to its weighted parameter of computational time and the required number of CPU cycles to accomplish the computation task I_n, i.e., the computation workload. Therefore, the MU with the higher computation workload will be assigned more computation resource by the MEC server. This is reasonable since multiple MUs share the same computation pool (at the MEC server) and each MU is of equal opportunity to offload its computation task and exploit powerful computation capabilities at the MEC server.

C. JOINT COMPUTATION OFFLOADING, BANDWIDTH, AND COMPUTATION RESOURCE ALLOCATION

1) ALGORITHMIC DETAILS

According to the analysis of the offloading decision and resource allocation discussed in previous two subsections, we propose an iterative algorithm to tackle the original optimization problem (6). The details of the proposed algorithm are summarized in Algorithm 1.

Algorithm 1 Algorithm of Joint Offloading decision, Bandwidth, and Computation resource Allocation (JOBCA).

```
Initialization: Select a random $\alpha$ and $\boldsymbol{f}$, and solve to get the initial solution $\boldsymbol{x}$, and the iteration $t = 0$.
repeat
    Set $t = t + 1$.
    The offloading decision phase: for a given $(\alpha, \boldsymbol{f})$
        Check all the offloading conditions to find $\mathcal{N}_{\text{off}}(t)$
        and $\mathcal{N}_{\text{rej}}(t)$.
        Check the backhaul constraint to find $\mathcal{N}_{\text{rem}}$.
        Update $\mathcal{N}_{\text{rej}}(t) := \mathcal{N}_{\text{rej}}(t) \cup \mathcal{N}_{\text{rem}}$ and $\mathcal{N}_{\text{off}}(t) := \mathcal{N}_{\text{off}}(t) \setminus \mathcal{N}_{\text{rem}}$.
    The resource allocation phase: for a given $\boldsymbol{x}$
        Solve $\frac{\partial f}{\partial \alpha} = 0$ and compare the results with the boundary values $0 \leq \alpha \leq \alpha_{\text{up}}$ to get $\alpha$.
        Update $\boldsymbol{f}$ is achieved using Eq. (12).
    Update $\boldsymbol{x}^*(t)$ according to (13).
    Update remote computation overheads of the MUs and the backhaul capacity availability.
until The set of offloading MUs $\mathcal{N}_{\text{off}}$ remains unchanged for two consecutive times.
Output: the optimal solution $(\boldsymbol{x}^*, \alpha^*, \boldsymbol{f}^*)$.
```

In Algorithm 1, the MUs that are rejected by the MEC server in the iteration t will be reconsidered as new MUs and will join the offloading decision problem (7) in the next iteration $(t+1)$. In order to start new iterations, local/remote computation overheads of the MUs and the backhaul capacity availability are updated in the last step of the previous iterations. Here, only the MUs that are accepted by the MEC server in the iteration t will join the resource allocation problem (8) in the iteration $(t+1)$. However, both the rejected and accepted MUs are considered in the offloading decision problem (7) so as to optimize the offloading decisions $\boldsymbol{x}^*(t+1)$, which are updated, as follows:

$$\boldsymbol{x}^*(t+1)=\begin{cases}\boldsymbol{x}^*(t), & \text{if } \sum_{n\in\mathcal{N}} Z_n(t)\le\sum_{n\in\mathcal{N}} Z_n(t+1)\\ \boldsymbol{x}(t+1), & \text{otherwise,}\end{cases} \tag{13}$$

where $\boldsymbol{x}(t+1)$ is the solution of the problem (7) for a given $(\alpha(t),\boldsymbol{f}(t))$. The algorithm will terminate once the set of offloading/rejected users does not change for two consecutive iterations. In other words, when there is no further computation offloading requests from MUs for two consecutive iterations, the algorithm will stop.

2) COMPLEXITY ANALYSIS

To obtain the solution, the original problem (6) is decomposed into two subproblems of the offloading decisions and joint computation resource and bandwidth partitioning. Specifically, the MUs with $Z_n^r \le Z_n^l$ are firstly assumed to be profitable from computation offloading and subsequently the MU with the smallest computation offloading gain, i.e., $\left(Z_n^l - Z_n^r\right)$, is removed from the offloading MU candidate list until the backhaul capacity constraint is not violated. Therefore, the complexity of finding the computation offloading decisions is $\mathcal{O}(N)$. The bandwidth partitioning factor is achieved by solving the first-order derivative of $f(\alpha)$ and comparing the result with the boundary values $\left[0\;\; \alpha_{\mathrm{up}}\right]$, hence the complexity $\mathcal{O}(1)$. Moreover, the computation resources for offloading MUs are obtained via the duality method. Let $\mathcal{T}$ be the number of iterations required to update the offloading decisions, bandwidth partitioning factor, and computation resources. For a given $\mathcal{T}$, the computational complexity of the proposed algorithm is $\mathcal{O}(\mathcal{T}N)$. Since we consider small cells with wireless backhaul, the number of MUs is often not large. As a result, the computational complexity of the proposed algorithm is affordable.

3) CONVERGENCE ANALYSIS

The JOBCA algorithm has two phases: the offloading decision phase and the resource (computation and bandwidth spectrum) allocation phase. Once the offloading decision phase terminates, all the MUs are either accepted or rejected by the MEC server to offload their computation tasks. The process of determining the offloading decisions is similar to the many-to-one matching game with the deferred-acceptance algorithm [20]. Hence, at each iteration t, the outputs of the offloading decision problem, $\mathcal{N}_{\mathrm{rej}}(t)$ and $\mathcal{N}_{\mathrm{off}}(t)$, are all stable, in which no MU in $\mathcal{N}_{\mathrm{rej}}(t)$ can join $\mathcal{N}_{\mathrm{off}}(t)$ and no MU in $\mathcal{N}_{\mathrm{off}}(t)$ can leave to join $\mathcal{N}_{\mathrm{rej}}(t)$. Moreover, the solution for the offloading decision phase meets the backhaul capacity constraint as well as reducing the system-wide computation overhead, i.e., $\sum_{n\in\mathcal{N}} Z_n(t) \ge \sum_{n\in\mathcal{N}} Z_n(t+1)$. Additionally, the feasible solution set is finite and reduced after each iteration, i.e., the feasible solution set of offloading MUs that both satisfy the backhaul capacity constraint and reduce the system-wide computation overhead becomes smaller after each iteration. As a result, the JOBCA algorithm is guaranteed to converge after a finite number of iterations.

V. FURTHER DISCUSSIONS AND EXTENSIONS

In this section, we first explain how to extend our proposed framework to multi-cell settings, then consider partial computation offloading, and finally discuss DRL for computation offloading in a dynamic MEC system with wireless backhaul.

A. COMPUTATION OFFLOADING IN ULTRA-DENSE NETWORKS

Due to the low transmit power of SBSs and the wall penetration loss, the effect of inter-cell interference for indoor small cells can be small when considering UDNs. However, for outdoor small cells, one major issue is to account for the strong inter-cell interference. Now, we consider an MEC system with one MEC server collocated with the MBS, J small cells deployed within the coverage of the macro cell, and N MUs. We assume that the MBS is equipped with a very large number of antennas and has perfect channel state information (i.e., it knows the channel gain matrix for all SBSs) and the MBS is assumed to employ the typical zero-forcing beamforming technique. The set of SBSs is denoted as $\mathcal{J} = \{1, 2, \ldots, J\}$ and the set of MUs in the j-th cell is presented by $\mathcal{I}_j$. It is assumed that the bandwidth resources allocated to the wireless access transmission is divided orthogonally into S subchannels, the set of subchannels is $\mathcal{S}$, and the resources $\mathcal{S}$ are reused among all SBSs. The channel gain between the i-th MU of the j-th small cell and the j'-th SBS on the s-th subchannel is denoted as $h_{i(j),j'}^s$. The rate of the i-th MU in the j-th small cell on the s-th subchannel can be computed according to the Shanon capacity formula as follows:

$$\begin{aligned} r_{ij}^s &= B_s \log_2\left(1+\gamma_{ij}^s\right)\\ &= B_s \log_2\left(1+\frac{p_{ij}^s h_{i(j),j}^s}{n_0+\sum\limits_{j'\neq j}\sum\limits_{k\in\mathcal{I}_{j'}} p_{kj'}^s h_{k(j'),j}^s}\right), \end{aligned} \tag{14}$$

where $B_s = \alpha W/S$ is the bandwidth of a subchannel, p_{ij}^s is the transmit power of the i-th MU in the j-th small cell on the s-th subchannel. Denote by $\boldsymbol{p}_{ij} = \left[p_{ij}^1, \ldots, p_{ij}^S\right]$ the transmit power vector of the i-th MU in the j-th small cell. Accordingly, the transmission rate between the i-th user and

j-th SBS is

$$r_{ij} = \sum_{s \in \mathcal{S}} B_s a_{ij}^s \log_2 \left(1 + \gamma_{ij}^s\right),$$

where a_{ij}^s is the assignment indicator, which is 1 if the subchannel s of the j-th cell is assigned to the i-th user and 0 otherwise.

According to [42], the transmission rate of the wireless backhaul uplink from the SBS j to the MBS is given by

$$R_j^{\text{bh}} = (1-\alpha)\, W \log_2 \left(1 + \frac{M}{M-N} \frac{\beta_j P_j}{n_0}\right), \tag{15}$$

where M is the number of MBS antennas, β_j models the geometric attenuation and shadow fading, which is usually known in advance and remains constant over many coherence time intervals, and P_j is the transmit power of the j-th SBS. From (15), if the number of MBS antennas grows without bound, the backhaul capacity of the j-th SBS can be represented as $R_j^{\text{bh}} = (1-\alpha)\, W \log_2 \left(1 + \beta_j P_j / n_0\right)$. To eliminate the interference from other small cells, other beamforming techniques such as maximum-ratio combining and minimum mean-squared error can alo be used in the massive MIMO systems [42].

To formulate the optimization problem similar to problem (6) in UDNs, we reuse the same notations as given in Section III while using i as the user index and adding some extra indexes, e.g., subchannel index s and cell index j. Let us define $\boldsymbol{a}_{ij} = \left[a_{ij}^1, \ldots, a_{ij}^S\right]$, $\boldsymbol{a} = \left[\boldsymbol{a}_{11}, \ldots, \boldsymbol{a}_{|\mathcal{I}_J|J}\right]$, $\boldsymbol{x} = \left[x_{11}, \ldots, x_{|\mathcal{I}_J|J}\right]$, and $\boldsymbol{f} = \left[f_{11}^r, \ldots, f_{|\mathcal{I}_J|J}^r\right]$. We also denote the set of offloading MUs in the j-th cell as $\mathcal{I}_j^{\text{off}}$. The computation offloading problem in UDNs can be formulated as follows:

$$\begin{aligned}
&\min_{\{\alpha, \boldsymbol{x}, \boldsymbol{a}, \boldsymbol{f}\}} \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_j} Z_{ij}(x_{ij}, a_{ij}, \alpha, f_{ij}^r) \\
&\text{s.t. C1'}: x_{ij} = \{0, 1\}, \quad \forall j \in \mathcal{J}, i \ \in \mathcal{I}_j \\
&\quad \text{C2'}: 0 \leq \alpha \leq 1, \\
&\quad \text{C3'}: \sum_{i \in \mathcal{I}_j^{\text{off}}} r_{ij} \leq R_j^{\text{bh}}, \quad \forall j \in \mathcal{J} \\
&\quad \text{C4'}: f_{ij}^r > 0, \quad \forall j \in \mathcal{J}, i \ \in \mathcal{I}_j^{\text{off}} \\
&\quad \text{C5'}: \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_j^{\text{off}}} f_{ij}^r \leq f_0, \\
&\quad \text{C6'}: \sum_{i \in \mathcal{I}_j^{\text{off}}} a_{ij}^s \leq 1, \quad \forall j \in \mathcal{J}, \quad \forall s \in \mathcal{S} \\
&\quad \text{C7'}: a_{ij}^s \in \{0, 1\}, \quad \forall j \in \mathcal{J}, \ \forall i \in \mathcal{I}_j^{\text{off}}, \ \forall s \in \mathcal{S}.
\end{aligned} \tag{16}$$

Compared with the optimization problem (6), there are an additional variable, i.e., the subchannel allocation vector $\boldsymbol{a}$, and two additional constraints, that are C6' and C7'. The constraint C6' ensures that each subchannel is assigned to at most one MU and C7' indicates the binary subchannel assignment variables.

In the following, we briefly explain how our proposed framework can be extended to consider general scenarios, where one MBS covers several SBSs. The main idea is still based on the problem decomposition technique. The original problem (16) is decomposed into three subproblems: (i) offloading decision, (ii) subchannel assignment, and (iii) resource allocation.

1) OFFLOADING DECISION

For a given $(\alpha, \boldsymbol{a}, \boldsymbol{f})$, the problem to optimize the offloading decisions can be formulated as

$$\begin{aligned}
&\min_{\boldsymbol{x}} \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_j} Z_{ij}(x_{ij}) \\
&\text{s.t. } x_{ij} = \{0, 1\}, \quad \forall j \in \mathcal{J}, i \ \in \mathcal{I}_j \\
&\quad \sum_{i \in \mathcal{I}_j^{\text{off}}} r_{ij} \leq R_j^{\text{bh}}, \quad \forall j \in \mathcal{J}.
\end{aligned} \tag{P1}$$

One can observe that the subproblem (P1) is fully decoupled across different SBSs, thus (P1) can be further decomposed into J SBS-level subproblems. Each SBS-level subproblem can be solved using our proposed method as presented in Subsection IV-A.

2) SUBCHANNEL ASSIGNMENT

Given a set $(\alpha, \boldsymbol{x}, \boldsymbol{f})$, the subchannel assignment subproblem is as follows:

$$\begin{aligned}
&\min_{\boldsymbol{a}} \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_j^{\text{off}}} Z_{ij}(a_{ij}) \\
&\text{s.t. } \sum_{i \in \mathcal{I}_j^{\text{off}}} a_{ij}^s \leq 1, \quad \forall j \in \mathcal{J}, \ \forall s \in \mathcal{S} \\
&\quad a_{ij}^s \in \{0, 1\}, \quad \forall j \in \mathcal{J}, \ \forall i \in \mathcal{I}_j^{\text{off}}, \ \forall s \in \mathcal{S}.
\end{aligned} \tag{P2}$$

Similarly, the subproblem (P2) is fully decomposable, i.e., it can be decomposed into J SBS-level subproblems, each subproblem is to allocate S subchannels to the offloading users so as to minimize the computation overhead. When $x_{ij} = 1$, $Z_{ij} = Z_{ij}^r$, which can be rewritten as follows:

$$\begin{aligned}
Z_{ij} &= \lambda_{ij}^t \left(t_{ij}^{\text{exe}} + \frac{D_{ij}}{r_{ij}} + \frac{D_{ij}}{(1-\alpha)\, W R_j^{\text{bh}}}\right) \\
&\quad + \lambda_{ij}^e \left(p_{ij} \frac{D_{ij}}{r_{ij}} + P_j \frac{D_{ij}}{(1-\alpha)\, W R_j^{\text{bh}}}\right) \\
&= \text{Const}\left(\alpha, x_{ij}, f_{ij}^r\right) + \frac{1}{\sum_{s \in \mathcal{S}} a_{ij}^s \frac{r_{ij}^s}{\left(\lambda_{ij}^t + p_{ij} \lambda_{ij}^e\right) D_{ij}}},
\end{aligned} \tag{19}$$

where $\text{Const}\left(\alpha, x_{ij}, f_{ij}^r\right)$ indicates a constant value for a given $\left(\alpha, x_{ij}, f_{ij}^r\right)$, which is independent of $\boldsymbol{a}_{ij}$. The j-th SBS-level subproblem for subchannel assignment can be formulated as

$$\begin{aligned} &\max_{\boldsymbol{a}} \sum_{i\in\mathcal{I}_j^{\text{off}}}\sum_{s\in\mathcal{S}} a_{ij}^s \frac{r_{ij}^s}{\left(\lambda_{ij}^t + p_{ij}\lambda_{ij}^e\right) D_{ij}} \\ &\text{s.t.} \sum_{i\in\mathcal{I}_j^{\text{off}}} a_{ij}^s \le 1, \quad \forall s\in\mathcal{S} \\ &\qquad a_{ij}^s \in \{0,1\}, \quad \forall i\in\mathcal{I}_j^{\text{off}}, \forall s\in\mathcal{S}. \qquad \text{(P2j)} \end{aligned}$$

We indicate by $i(j,s)$ the MU served by the j-th SBS on the s-th subchannel. From (19), the solution to the subproblem (P2j) can be written as

$$\hat{i}(j,s) = \underset{i\in\mathcal{I}_j^{\text{off}}}{\operatorname{argmax}} \frac{r_{ij}^s}{\left(\lambda_{ij}^t + p_{ij}\lambda_{ij}^e\right) D_{ij}}.$$

3) RESOURCE ALLOCATION

Once the offloading decision and subchannel assignment vectors are obtained, they can be used for the joint optimization of wireless bandwidth and computing resource allocation as follows:

$$\begin{aligned} &\min_{\{\alpha,\boldsymbol{f}\}} \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}_j^{\text{off}}} Z_{ij}(\alpha, f_{ij}^r) \\ &\text{s.t. C2', C3', C4', C5', C6'}. \qquad \text{(P3)} \end{aligned}$$

As presented in Subsection IV-B, (P3) can be decomposed into two smaller subproblems: one for computing resource allocation at the MEC server and the other for wireless backhaul bandwidth allocation. After some algebraic manipulation, the allocation of computing resources for offloading users can be expressed as

$$f_{ij}^{r*} = \frac{f_0\sqrt{\lambda_{ij}^t C_{ij}}}{\sum\limits_{j\in\mathcal{J}, i\,\in\mathcal{I}_j^{\text{off}}} \sqrt{\lambda_{ij}^t C_{ij}}}$$

and the bandwidth partitioning factor is achieved by solving (10) with $t_{ij}^{'\text{ac}} = t_{ij}^{\text{ac}}\alpha$, $t_{ij}^{'\text{bh}} = t_{ij}^{\text{bh}}(1-\alpha)$, $E_{ij}^{'\text{ac}} = E_{ij}^{\text{ac}}\alpha$, and $E_{ij}^{'\text{bh}} = E_{ij}^{\text{bh}}(1-\alpha)$, $A = \sum_{j\in\mathcal{J}, i\,\in\mathcal{I}_j^{\text{off}}}\left(\lambda_{ij}^t t_{ij}^{'\text{ac}} + \lambda_{ij}^e E_{ij}^{'\text{ac}}\right)$, $B = \sum_{j\in\mathcal{J}, i\,\in\mathcal{I}_j^{\text{off}}}\left(\lambda_{ij}^t t_{ij}^{'\text{bh}} + \lambda_{ij}^e E_{ij}^{'\text{bh}}\right)$, and

$$\alpha_{\text{up}} = \min_{j\in\mathcal{J}}\left\{\frac{R_j^{\text{bh}}}{R_j^{\text{bh}} + \sum_{i\in\mathcal{I}_j^{\text{off}}}\frac{1}{S}\log_2\left(1+r_{ij}\right)}\right\}.$$

As can be seen from (14), the inter-cell interference must be estimated and measured at the SBSs and MBS as well. Some common approaches for inter-cell interference mitigation are power control and fractional frequency reuse. In the former approach, transmit powers of MUs and SBSs are controlled to minimize the computation overhead while satisfying their power budget. The latter approach groups nearby small cells into a cluster and divides the system bandwidth into spatial regions, each region corresponds to a cluster. Therefore, co-cluster small cells are assigned with different frequency bands. This technique can significantly mitigate the effect of co-cluster inter-cell interference while the inter-cluster interference is negligible and can be ignored. A design of computation offloading and resource allocation (transmit power and computation) in UDNs with wireless backhaul will be investigated in our future work.

Another important issue in UDNs is user association, where each MU must either associate with one among multiple small cells or simultaneously connect to multiple small cells [20]. In practice, SBSs connect to the MBS by distinct backhaul links, i.e., wired and wireless links. For small cells with wireless backhaul, beside the bandwidth partitioning factor, the transmit power strongly impacts the backhaul capacity. Therefore, each offloading MU needs to select the best SBS to associate with considering the quotas of different SBSs and the MBS, channel quality, and backhaul condition. Take single-user networks as an example, MUs would definitely connect to the SBS with the wired backhaul and best quality channel and then offload directly to the MEC server rather than offloading to the MEC server through a small cell with wireless backhaul. A joint consideration of user association, computation offloading, and resource allocation in UDNs with wireless backhaul is a promising direction which will be studied our future work.

B. PARTIAL OFFLOADING DECISION

In Subsection IV-A, the offloading decisions are found in a heuristic manner. This scheme is simple to implement and efficient in the case where a large number of MUs connected to an SBS. According to [20] and [43], there can be few MUs per small cell in UDHNs. For a given offloading decision vector $\boldsymbol{x}$, we only need to examine the backhaul capacity constraint and then, among feasible solutions, the offloading decision with the smallest computation overhead is selected as the optimal offloading decision. Therefore, in small cells, it is computationally efficient to achieve the optimal offloading solution by the exhaustive search.

We can obtain the lower bound in terms of the computational time and energy consumption of the original problem by considering the offloading decision relaxation of the problem (7) as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}} \textstyle\sum_{n\in\mathcal{N}} x_n\left(Z_n^r - Z_n^l\right) \\ &\text{s.t. C6}: x_n \in [0,1], \quad \forall n\in\mathcal{N} \\ &\quad\ \text{C7}: \textstyle\sum_{n\in\mathcal{N}_{\text{off}}} r_n\left(\mathcal{N}_{\text{off}}\right) \le R_{\text{bh}}. \qquad (21) \end{aligned}$$

Different to the problem (7), C1 is relaxed to become C6. In fact, this corresponds to partial offloading, i.e., a fraction of a computation task is offloaded to the MEC server while the remaining part is handled locally. Once the set of offloading MUs is given, both the left-hand-side and right-hand-side (RHS) of C7 are fixed, and then its feasibility can be checked easily. Moreover, the objective function is linear in and the RHS of C7 is independent of the offloading decisions $\boldsymbol{x}$. Hence, the final offloading decisions of MUs are either local handling or remote processing. As a result, the optimal solution of the relaxed problem is the same as that achieved by the exhaustive search. However, the relaxed optimization problem (21) becomes greatly complicated in UDHNs since each MU can divide its computation task to multiple subtasks and offload each computation subtask to the MEC server through a distinct SBS.

C. MACHINE LEARNING BASED COMPUTATION OFFLOADING

Recently, machine learning (ML) has emerged as an effective method for handling many challenges and problems in wireless and communication networks. In terms of computation offloading for MEC systems, there have been some works using ML, especially reinforcement learning (RL) for computation offloading problems, for example, [44], [45]. Two advantages of using RL for computation offloading problems are: (1) it is suitable for random and time-varying MEC systems and (2) it enables learning without a priori knowledge of network statistics [46].

In the current work, we consider a quasi-static network. To make our proposed algorithm feasible for MEC systems with fast-fading channel, one potential extension is applying DRL to solve the underlying optimization problem. In particular, the state, action, and reward of the RL agent can be modeled as follows:

- **State**: the system state is defined as the channel gains between MUs and SBS and between SBS and MBS, i.e., $s = \{\boldsymbol{h}, h_0\}$, where $\boldsymbol{h} = \{h_1, \ldots, h_N\}$.
- **Action**: the action consists of offloading decision $\boldsymbol{x}$ and resource allocation $(\alpha, \boldsymbol{f})$. Thus, the action vector can be given as $a = [\boldsymbol{x}, \alpha, \boldsymbol{f}]$.
- **Reward**: the objective of the original problem is to minimize the system-wide computation overhead while the goal of RL is to maximize the long-term reward. Therefore, the reward can be given as $\sum_{n\in\mathcal{N}} \left(Z_n^l - Z_n^r\,(s, a\,)\right)$.

The RL-based problem can be solved to obtain the optimal policy by using conventional methods such as dynamic programming and Q-learning. To deal with the curse of dimensionality due to the large number of MUs, DRL-based methods can be used instead, for example, deep Q network (DQN) and dueling DQN. DRL-based methods can also be used to extend our work to consider a more complex MEC system, which takes interference into consideration.

VI. NUMERICAL RESULTS

A. SIMULATION SETTINGS

In this section, we will demonstrate the performance of the proposed algorithm (JODBA) through numerical study. Consider an MEC system with an MBS and an SBS, which have the coverage radius of 250 m and 50 m, respectively. The SBS (MUs) is randomly positioned within the coverage of the MBS (SBS) and the minimal distance from the MBS to SBS is 40 m. The system bandwidth is 20 MHz, AWGN power is -100 dBm, and the transmit power of both the SBS and MUs is 100 mW. The pathloss model is assumed to follow the log-distance path loss model [47], where the MBS-to-SBS path loss for the distance r is calculated as $L(r) = 15.3 + 37.6\log_{10}(r)$ and the SBS-to-MU path loss for distance r is computed as $L(r) = 38.46 + 20\log_{10}(r)$. The large-scale shadowing is modeled by a log-normal distribution with zero mean and standard deviation 8 dB and the small-scale fading coefficients are assumed to be Rayleigh random variables with unit variances.

Regarding the computation model, the face recognition application is adopted, where the computation input data size is 420 KB and the total required number of CPU cycles is 1000 Megacycles [18]. The CPU computational capability f_n^l of MU n is randomly assigned from the set $\{0.5, 0.8, 1.0\}$ GHz and the maximum computation resource at the MEC server f_0 is 10 GHz. The weighted parameters of computational time and energy consumption are both 0.5, i.e., $\lambda_n^t = \lambda_n^e = 0.5, \forall n \in \mathcal{N}$. Simulation results are obtained with 5000 channel realizations on average and MUs, SBS, and MBS locations are uniformly distributed randomly in each realization.

B. SIMULATION RESULTS

For performance evaluation, two benchmark schemes are considered and compared with our proposed algorithm:

1) *Local computing only*: All MUs perform their computations locally, i.e., $x_n = 0, \forall n \in \mathcal{N}$.
2) *Offloading only*: All MUs offload their computations to the MEC server, which will execute all the tasks from MUs, i.e., $x_n = 1, \forall n \in \mathcal{N}$.

Fig. 3 shows the variation and convergence of the system-wide computation overhead (i.e., total computation overhead) for Algorithm 1 versus the number of iterations in two scenarios: 10 and 14 mobile users. It is observed that the larger the number of MUs is, the higher the system-wide computation overhead is generated. In addition, the proposed algorithm can converge to the stable solution within ten iterations. This result, together with the previous convergence analysis, confirms that the proposed iterative algorithm is convergent.

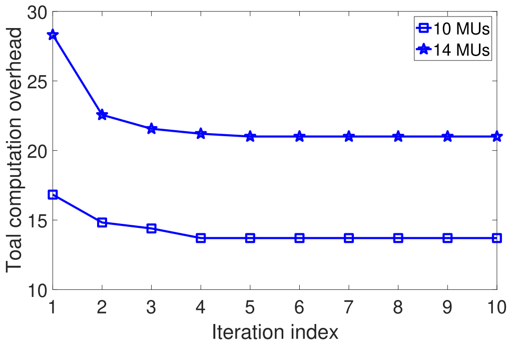
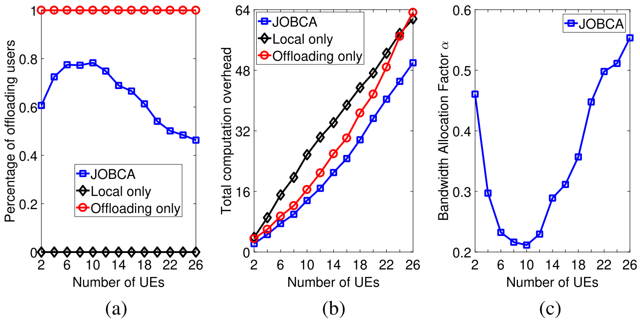


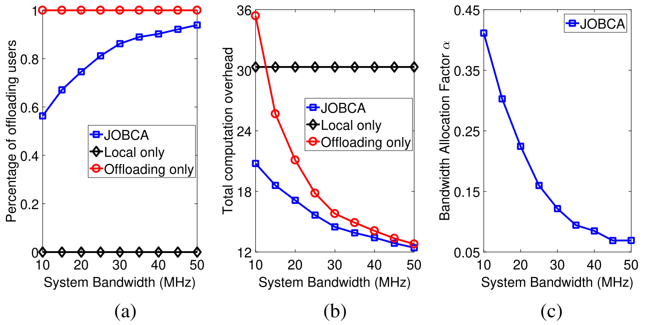
FIGURE 3. The convergence in terms of the system-wide computation overhead versus the number of iterations.

In the second experiment, we vary the number of MUs and observe the various performance of the proposed algorithm



(a) (b) (c)

FIGURE 4. Comparison of JOBCA and two baseline schemes under different numbers of MUs. (a) Percentage of offloading UEs. (b) Computation overhead. (c) Bandwidth Allocation Factor.



(a) (b) (c)

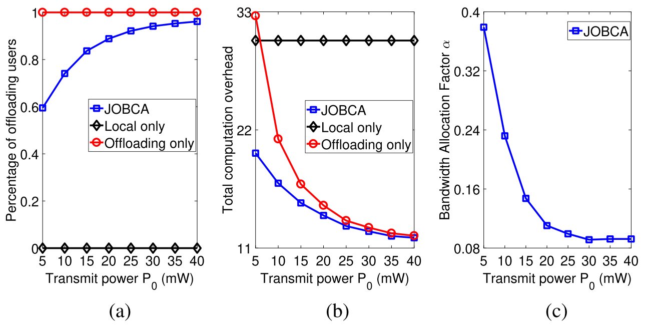
FIGURE 5. Performance comparison under different system bandwidths. (a) Percentage of offloading MUs. (b) Computation overhead. (c) Bandwidth Allocation Factor.

as well as the benchmark schemes. From Fig. (4a), the percentage of offloading MUs of the local and offloading only schemes are respectively 0 and 1, and that of JOBCA changes with the number of MUs. When the number of MUs is small, the percentage of offloading MUs keeps increasing. This is because when the number of MUs is sufficiently small, each MU has a high opportunity to be assigned large computation resource by the MEC server, and then the remote computation overhead is lower than the local one. However, when the number of MUs becomes large enough, the percentage of offloading MUs begins to decrease. This is reasonable since more MUs tend to offload their computations to the MEC server and the computation resource assigned to each offloading MU becomes smaller. Therefore, the MEC server rejects some of the requested MUs that incur higher computation overhead compared to the local computing scheme.

It can be observed from Fig. (4b) that JOBCA can achieve relatively lower computation overhead compared with two baseline schemes (local and offloading only). This is due to the fact that the offloading decision, bandwidth, and computation resource are jointly optimized in our proposed framework such that computation offloading is advantageous to offloading MUs. Fig. (4b) additionally reveals that the offloading only scheme can generate higher computation overhead than the local computing only approach when the number of MUs becomes sufficiently large. It is due to the competition among MUs for the limited computation resource at the MEC server. Fig. (4c) shows the bandwidth allocation factor α w.r.t. the number of MUs. It can be seen that the bandwidth factor is inversely proportional to the percentage of offloading MUs, as illustrated in Fig. (4a). The reason for this is that at first the wireless access rate increases with the number of offloading MUs and in turn the larger backhaul capacity, i.e., the lower α value according to formulas of r_n and R_{bh} in Subsection III-A, is required to make the backhaul capacity constraint satisfied. Nevertheless, when the wireless access rate declines, a larger portion of bandwidth α can be allocated for the wireless access transmission, i.e., $(1-\alpha)$ fraction for wireless backhaul is smaller.

In Fig. 5, we plot the simulation results as functions of the system bandwidth, where $N = 12$ (MUs) and $f_0 = 10$ (GHz). We can see from Figs. (5a) and (5c) that as the wireless channel bandwidth increases, the percentage of offloading users and bandwidth partitioning factor increases and decreases, respectively. In fact, increasing the system bandwidth provides MUs with higher opportunities to reduce the offloading time and to offload computation tasks to the MEC server. Accordingly, the bandwidth partitioning factor α is a decreasing function of the wireless channel bandwidth as well. It is reported by Fig. (5b) that when the system bandwidth is small, offloading all the computation tasks to the MEC server is not efficient. It is due to the fact that the offloading time decreases as the system bandwidth increases. Therefore, the offloading only scheme is more beneficial than the local only scheme if and only if the system bandwidth is large enough (15 MHz in this simulation setting). We can again observe that the proposed algorithm JOBCA generates significantly lower overhead compared to the two baseline schemes. However, when the system bandwidth is relatively large, e.g., 50 MHz, the percentage of offloading users by our proposed algorithm can nearly reach the maximal point (100% offloading users) and the computation overhead by the offloading only scheme is as low as that by our proposed algorithm. This is reasonable since the computation offloading time is quite small when the system bandwidth is sufficiently large.

In Fig. 6, we compare the performance of the three schemes when the transmit power of the SBS P_0 is varied. As can be

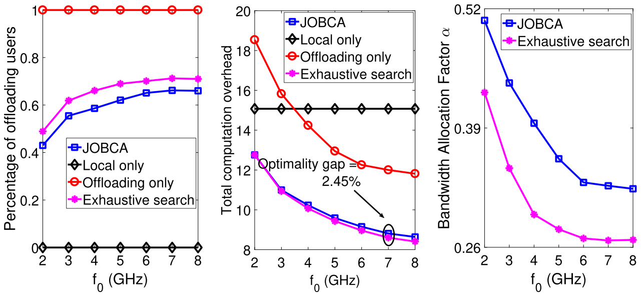


(a) (b) (c)

FIGURE 6. Performance comparison under variant transmit power of the SBS P_0. (a) Percentage of offloading MUs. (b) Computation overhead. (c) Bandwidth Allocation Factor.

seen, the performance trends in three sub-figures, i.e., the percentage of offloading users, system-wide computation overhead, and bandwidth partitioning factor, are similar to those in Fig. 5. The impacts of the system bandwidth W and transmit power P_0 are similar in the sense that the offloading time is a decreasing function of both parameters. It is worth noting that due to the short distance between MUs and the SBS, the channel gains h_n are usually much larger than h_0. In addition, the power budget of the SBS is higher than those of MUs. Therefore, as the result of formulae (2), (3), (4), (5), the transmit power P_0 has stronger impact on the system performance than p_n and it is important to efficiently allocate the transmit powers of SAPs to determine the offloading decisions of MUs.

In the final experiment, we examine the impacts of the maximum computational capability f_0 on the performance. Fig. (7a) shows the percentage of offloading MUs w.r.t. the increasing computational capability f_0, where there are 6 MUs. As we can see, the percentage of offloading MUs increases with the increasing of f_0. The reason is that, when the maximum computational capability f_0 increases, more computation resource can be allocated to each offloading MU (in this case, the number of MUs is fixed). Therefore, remote computation overhead of offloading MUs tends to decline as f_0 increases. The result from Fig. (7c) reports that the higher the maximum computational capability f_0, the lower the bandwidth allocation factor α. The observations from Figs. (7a)-(7c) identically match up with those in Fig. 4, i.e., the percentage of offloading MUs increases in inverse proportion to the bandwidth allocation factor and vice versa. As depicted in Fig. (7b), when the maximum computational capability f_0 is small enough ($f_0 < 5$ GHz in this case), locally performing computations is better than offloading to the remote MEC server.



(a) (b) (c)

FIGURE 7. Performance comparison under different f_0. (a) Percentage of offloading MUs. (b) Computation overhead. (c) Bandwidth Allocation Factor.

In order to evaluate the optimality gap of the proposed algorithm, we further compare JOBCA with the exhaustive search solution, where the bandwidth partitioning factor and computation resource are jointly optimized for all of the feasible offloading solutions and then the one with the lowest system-wide computation overhead is selected as the optimal solution. It is worth noting that the exhaustive search is only affordable for the networks with few MUs, for example, 4 cellular users in small cell networks [20], [43]. For general scenarios with a massive number of IoT devices connected with the SBS to handle their computations, the exhaustive search will lead to extremely high computational complexity and long processing time.

Fig. 7 confirms the exhaustive search achieves the best performance among the schemes since more MUs can benefit from computation offloading (a larger portion of the system bandwidth is allocated for wireless backhaul transmission, i.e., smaller α). It is observed from Fig. (7b) that the optimality gap between our proposed algorithm and the exhaustive search generally increases with the maximum computational capability f_0 at the MEC server. It is reasonable since our proposed algorithm may reject to handle computations of some MUs who can indeed benefit from computation offloading with the increment of the maximum computational capability f_0, while the exhaustive search finds the optimal solution by selecting the one with the smallest computation overhead. Another observation from Fig. (7b) is that at $f_0 = 7$ GHz and 6 MUs, our proposed algorithm generates the system-wide computation overhead of 8.8117, which is close to that of the exhaustive search and our proposed scheme achieves the optimality gap of 2.45%, and 36.19 % and 71.08 % lower than those of the offloading only and local only schemes, respectively.

VII. CONCLUSION

In this paper, we studied the computation offloading problem in mobile edge computing with wireless backhaul. A joint problem of task offloading, wireless backhaul bandwidth partitioning, and computation resource allocation was investigated. Since the original problem is hard to tackle, we decomposed it into subproblems of offloading decision and joint backhaul bandwidth and computation resource allocation, which are solved individually and iteratively. We then proposed an algorithm JOBCA. We conducted numerical studies to analyze our proposed algorithm and two baseline schemes under different values of the maximum computational capability, number of MUs, system bandwidth, and transmit power of the SBS. These numerical studies validated that our proposed algorithm can improve significantly the network performance compared with two baseline solutions in terms of the system-wide computation overhead and the number of offloading users. In addition, our proposed algorithm could perform close to that of the centralized exhaustive search with the small optimality gap of 2.45% at the maximum computational capability $f_0 = 7$ GHz.

Our work can provide a useful guideline for mobile operators when they deliver MEC services to MUs. Based on the available spectrum, the number of expected users, channel qualities, and hosted applications, the mobile operators can utilize our framework to partition the bandwidth spectrum between wireless access and backhaul transmissions, and allocate the computing resources to serve the offloading users. For example, some users in a rural area need to use

MEC services; however, they/their SBS would not be able to directly connect to the MEC server. In this case, applying the solution with wireless backhaul between the SBS and MBS is technically feasible and suitable. In addition, the designs and results in this paper motivate researchers to further develop complex frameworks of computation offloading and resource allocation in MEC systems, for example, extension to dense HetNets, consideration of mixed wireless and wired backhaul links, and distributed offloading decisions in multi-users MEC HetNets.

REFERENCES

[1] Q. V. Pham and W. J. Hwang, "Energy-efficient power control in uplink spectrum-sharing heterogeneous networks," *Int. J. Commun. Syst.*, vol. 31, no. 14, p. e3717, Sep. 2018.

[2] X. Ge, H. Cheng, M. Guizani, and T. Han, "5G wireless backhaul networks: Challenges and research advances," *IEEE Netw.*, vol. 28, no. 6, pp. 6–11, Nov. 2014.

[3] U. Siddique, H. Tabassum, E. Hossain, and D. I. Kim, "Wireless backhauling of 5G small cells: Challenges and solution approaches," *IEEE Wireless Commun.*, vol. 22, no. 5, pp. 22–31, Oct. 2015.

[4] T. M. Nguyen, A. Yadav, W. Ajib, and C. Assi, "Resource allocation in two-tier wireless backhaul heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6690–6704, Oct. 2016.

[5] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generat. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.

[6] ETSI. (2017). *Multi-Access Edge Computing*. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/multi-access-edg%e-computing

[7] 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, 4th Quart., 2017.

[8] 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, 3rd Quart., 2017.

[9] T. M. Nguyen, A. Yadav, W. Ajib, and C. Assi, "Centralized and distributed energy efficiency designs in wireless backhaul HetNets," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4711–4726, Jul. 2017.

[10] Y. Liu, L. Lu, G. Y. Li, Q. Cui, and W. Han, "Joint user association and spectrum allocation for small cell networks with wireless backhauls," *IEEE Wireless Commun. Lett.*, vol. 5, no. 5, pp. 496–499, Oct. 2016.

[11] G. Nie, H. Tian, C. Sengul, and P. Zhang, "Forward and backhaul link optimization for energy efficient OFDMA small cell networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1080–1093, Feb. 2017.

[12] H. Zhang, H. Liu, J. Cheng, and V. C. M. Leung, "Downlink energy efficiency of power allocation and wireless backhaul bandwidth allocation in heterogeneous small cell networks," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1705–1716, Apr. 2018.

[13] Q. Han, B. Yang, G. Miao, C. Chen, X. Wang, and X. Guan, "Backhaul-aware user association and resource allocation for energy-constrained HetNets," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 580–593, Jan. 2017.

[14] 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.

[15] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[16] 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.

[17] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[18] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[19] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, May 2018.

[20] Q. V. Pham, T. LeAnh, N. H. Tran, and C. S. Hong, "Decentralized computation offoading and resource allocation for mobile-edge computing: A matching game approach," *IEEE Access*, to be published, doi: 10.1109/ACCESS.2018.2882800.

[21] Y. Wu *et al.*, "Secrecy-driven resource management for vehicular computation offloading networks," *IEEE Netw.*, vol. 32, no. 3, pp. 84–91, May 2018.

[22] J. Zheng, Y. Cai, Y. Wu, and X. S. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. Mobile Comput.*, to be published.

[23] T. Brummett, P. Sheinidashtegol, D. Sarkar, and M. Galloway, "Performance metrics of local cloud computing architectures," in *Proc. IEEE 2nd Int. Conf. Cyber Secur. Cloud Comput.*, Nov. 2015, pp. 25–30.

[24] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 725–737, Oct. 2017.

[25] 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.

[26] (Sep. 2014). *Mobile-Edge Computing: Introductory Technical White Paper*. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf

[27] C. F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *Proc. IEEE Globecom Workshops (GC WKSHPS)*, Dec. 2017, pp. 1–7.

[28] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[29] Y. Mao, J. Zhang, Z. Chen, 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.

[30] W. Hao and S. Yang, "Small cell cluster-based resource allocation for wireless backhaul in two-tier heterogeneous networks with massive MIMO," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 509–523, Jan. 2018.

[31] M. Fiorani, S. Tombaz, P. Monti, M. Casoni, and L. Wosinska, "Green backhauling for rural areas," in *Proc. Int. Conf. Opt. Netw. Design Modeling*, May 2014, pp. 114–119.

[32] E. K. Markakis *et al.*, "Efficient next generation emergency communications over multi-access edge computing," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 92–97, Nov. 2017.

[33] N. Omidvar, A. Liu, V. Lau, F. Zhang, D. H. K. Tsang, and M. R. Pakravan, "Optimal hierarchical radio resource management for HetNets with flexible backhaul," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4239–4255, Jul. 2018.

[34] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2017.

[35] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[36] Q.-V. Pham, H.-L. To, and W.-J. Hwang, "A multi-timescale cross-layer approach for wireless ad hoc networks," *Comput. Netw.*, vol. 91, no. 11, pp. 471–482, 2015.

[37] J. Wallenius, J. S. Dyer, P. C. Fishburn, R. E. Steuer, S. Zionts, and K. Deb, "Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead," *Manage. Sci.*, vol. 54, no. 7, pp. 1336–1349, Jul. 2008.

[38] Q. V. Pham and W. J. Hwang, "Fairness-aware spectral and energy efficiency in spectrum-sharing wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10207–10219, Nov. 2017.

[39] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming*, vol. 233. New York, NY, USA: Springer, 2006.

[40] Q.-V. Pham and W.-J. Hwang, "Resource allocation for heterogeneous traffic in complex communication networks," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 10, pp. 959–963, Oct. 2016.

[41] Q.-V. Pham and W.-J. Hwang, "Network utility maximization-based congestion control over wireless networks: A survey and potential directives," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1173–1200, 2nd Quart., 2017.

[42] 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.

[43] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, "Femtocell networks: A survey," *IEEE Commun. Mag.*, vol. 46, no. 9, pp. 59–67, Sep. 2008.

[44] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for mec," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[45] H. Cao and J. Cai, "Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 752–764, Jan. 2018.

[46] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2018.2876279.

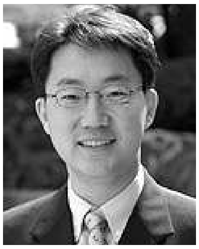
[47] H. Wang and Z. Ding, "Power control and resource allocation for outage balancing in femtocell networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 2043–2057, Apr. 2015.

QUOC-VIET PHAM (M'18) received the B.S. degree in electronics and telecommunications engineering from the Hanoi University of Science and Technology, Vietnam, in 2013, and the M.S. and Ph.D. degrees in telecommunications engineering from Inje University, South Korea, in 2015 and 2017, respectively. From 2017 to 2018, he was a Post-Doctoral Researcher at Kyung Hee University, South Korea. He is currently a Research Professor at the ICT Convergence Center, Changwon National University, South Korea. His research interests include network optimization, mobile edge/cloud computing, and resource allocation for wireless networks. He received the Best Ph.D. Thesis Award in engineering from Inje University in 2017.

LONG BAO LE (S'04–M'07–SM'12) received the B.Eng. degree in electrical engineering from the Ho Chi Minh City University of Technology, Vietnam, in 1999, the M.Eng. degree in telecommunications from the Asian Institute of Technology, Thailand, in 2002, and the Ph.D. degree in electrical engineering from the University of Manitoba, Canada, in 2007. He was a Post-Doctoral Researcher with the Massachusetts Institute of Technology from 2008 to 2010 and with the University of Waterloo from 2007 to 2008. Since 2010, he has been with the Institut National de la Recherche Scientifique (INRS), Universite du Quebec, Montreal, QC, Canada, where he is currently an Associate Professor. He has co-authored the books *Radio Resource Management in Multi-Tier Cellular Wireless Networks* (Wiley, 2013) and *Radio Resource Management in Wireless Networks: An Engineering Approach* (Cambridge University Press, 2017). His current research interests include smart grids, cognitive radio, radio resource management, network control and optimization, and emerging enabling technologies for 5G wireless systems. He is currently a member of the Editorial Board of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.

SANG-HWA CHUNG received the B.S. degree in electrical engineering from Seoul National University in 1985, the M.S. degree in computer engineering from Iowa State University in 1988, and the Ph.D. degree in computer engineering from the University of Southern California in 1993.

He was an Assistant Professor with the Electrical and Computer Engineering Department, University of Central Florida, from 1993 to 1994. He is currently a Professor with the Computer Engineering Department, Pusan National University, South Korea. He also serves as the Director of the Dong-Nam Grand ICT R&D Center, South Korea. His research interests are in the areas of sensor networks, embedded systems, fog computing, and IoT.



WON-JOO HWANG (S'01–M'03–SM'17) received the B.S. and M.S. degrees in computer engineering from Pusan National University, Pusan, South Korea, in 1998 and 2000, respectively, and the Ph.D. degree in information systems engineering from Osaka University, Osaka, Japan, in 2002. He is currently a Full Professor at Inje University, Gimhae, South Korea. His research interests include network optimization and cross layer design.