IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 9, 2020, accepted March 5, 2020, date of publication March 10, 2020, date of current version March 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979774

Computation Offloading and Resource Allocation for the Internet of Things in Energy-Constrained MEC-Enabled HetNets

LIANGRUI TANG AND HAILIN HU

State Key Laboratory of Alternate Electrical Power System With Renewable Energy Sources, North China Electric Power University, Beijing 102206, China

Corresponding author: Hailin Hu (1172101012@ncepu.edu.cn)

This work was supported in part by the National High Technology Research and Development of China 863 Program under Grant 2014AA01A701, in part by the Beijing Natural Science Foundation under Grant 414142049, in part by the National Natural Science Foundation of China under Grant 51507063, and in part by the Fundamental Research Funds for the Central Universities under Grant 2019QN103.

ABSTRACT In this paper, we present our investigation of a latency-minimization offloading problem for internet of things (IoT) terminals in multiple access edge computing (MEC)-enabled heterogeneous networks (HetNets), which jointly optimizes computation and communication resource allocation. Different from related works, the inter-user interferences caused by computation offloading demonstrate effective management in this paper. We also consider the limited battery capacity for IoT terminals for an energy-limited network. Then, we formulate a joint computation offloading and resource allocation optimization problem to minimize the weight-sum delay of users under the constraint of inter-user interference and energy consumption. Since the problem we formulated is a mixed integer non-linear programming (MINLP) problem, the optimal solution can't be easily obtained. Thus, we decompose the problem into multiple sub-problems. First, we obtain the optimal close solution for local CPU frequencies for each user. Then we propose a low complexity algorithm by using the CVX tool and the successive convex approximation approach (SCA). Finally, we propose a distributed computation offloading algorithm. The simulation results compare the performance of the proposed offloading scheme with different algorithms. We also analyze the influence of network parameters on the network latency and obtain some interesting conclusions.

INDEX TERMS MEC enabled HetNets, computation offloading, resource allocation, Internet of Things.

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

Over the past few years, more and more smart mobile devices and computing-intensive applications have accelerated the development of the internet of things (IoT) [1]. Due to the exponential growth of mobile data traffic, merely relying on the traditional cloud computing is not enough to the latency requirements of IoT devices and applications, which may also cause a severe computation and communication load on the cloud computing center. Consequently, the emergence of *multi-access edge computing* (MEC) has been considered to be a promising solution to this challenge. MEC offers the computation capability for offloading computation tasks from mobile devices and applications at the edge of the network [2], [3]. Moreover, MEC technology can be combined with *heterogeneous networks* (HetNets). Each base station (BS) is placed by a MEC sever to execute offloading tasks from its accessed mobile devices in HetNets, which has enabled the creation of new, potential network architecture in the 5G/B5G communication era [4], [5]. Thus, the end to end user experience and overall spectrum efficiency of the network can be further improved, simultaneously.

However, the task offloading decision and resource allocation for IoT devices are a key research problem in MEC systems [6], [7]. In MEC systems, IoT users (UEs) may choose to offload their generated tasks to the edge server for execution due to the inherently limited computing capacity and battery power of MEC systems. This may result in a large end-to-end delay for UEs. Furthermore, it may not be able to serve all UEs that have limited wireless and computation capacity for the edge server. Therefore, it is essential to propose an appropriate task offloading decision and resource allocation method for multi-users based on the system revenue and UEs'

The associate editor coordinating the review of this manuscript and approving it for publication was Shree Krishna Sharma.

quality of services (QoS) requirements of mobile devices [8]. Specifically, the offloading decision and resources occupied by IoT devices need to be optimized by considering the local computing capacity, the channel environment, the energy consumption, and the task execution delay requirement.

Different from traditional MEC systems, in the MEC-enabled HetNets, the task offloading decision may cause inter-user interference in the process of task uploading to the MEC server. And the interference may become more severe and complicated especially in the case of dense deployment of small BSs (SBSs). Interference from other UEs in the same layer and cross-layer will exist in the network. Therefore, the computation offloading decision and resource allocation problem in MEC-enabled HetNets must consider the effective interference management form other UEs. The energy-limited characteristic of IoT devices should also be considered when addressing the research problem. In this paper, we consider the interference and energy constraints in the process of task execution, and we focus on proposing an appropriate computation offloading and resource allocation method.

B. RELATED WORKS

In recent years, the problem of computation offloading and resource allocation has attracted the attention of many researchers. Most of the related works focus on the single-user scenario [9]–[11] and the multi-user scenario. In [12], the authors propose a partial offloading decision and resource allocation algorithm to minimize UEs' energy consumption in multi-user time division multiple access (TDMA) and frequency division multiple access (FDMA) MEC systems. In [13], the authors consider an MEC system combined with non-orthogonal multiple access (NOMA) technology (NOMA-MEC) for multi-users' task uploading and downloading. Then, the total system energy consumption is minimized by optimizing the transmit power, transmission time allocation, and task offloading decision. In [14], the authors optimize the task offloading decisions by minimizing the sum of all UEs' delay in the multi-user NOMA-MEC system. In [15], the authors combine the multi-user NOMA-MEC system with multi-antenna technology for task offloading. Then, different UEs can offload their computing tasks to the multi-antenna BS at the same time and with the same frequency resource. Each BS can also use the Successive Interference Cancellation (SIC) method to effectively decode and perform the UEs' offloading tasks. In [16], the authors minimize the network energy consumption for the multi-user system by jointly optimizing the UEs' task offloading decision, spectrum, power, and computing resource allocation in a densely deployed small cell network scenario. However, these studies did not consider the problem of interference avoidance between UEs.

The execution latency and energy consumption are usually two important metrics for the evaluation of computation offloading and resource allocation. In [17], all UEs' offloading decision and resource allocation are optimized to minimize the end-to-end delay of the UEs for the multi-user TDMA MEC system, but the UEs' energy consumption is not considered. In [18], the method of UEs' task offloading and multi-user scheduling for NB-IoT edge computing system is proposed to minimize the long-term average weighted sum of UEs' delay and energy consumption under the assumption of random traffic arrival, and Markov decision process model is adopted. In [19], the authors focus on the trade-off between the UEs' delay and energy consumption in the problem of joint optimization of UEs' task offloading decision, communication and computing resource allocation and the residual energy of the terminal is introduced as the weight. In [20], the authors propose a joint management method for online wireless and computing resources for multi-user MEC systems to minimize the long-term average weighted power consumption of mobile devices and MEC servers. In [21], all UEs' energy consumption is minimized with the constraint of execution delay by optimizing task allocation offloading decision, wireless and computing resources allocation for UEs in the NOMA-MEC system.

Different from the related works, we deploy the MEC system in HetNets and consider the problem of inter-user interference management during task offloading. Finally, we simultaneously optimize the UEs' task offloading decision, computation, and communication resource allocation. While execution latency is very important to UEs in IoT, especially for latency-sensitive UEs, the amount of energy for each IoT device is limited. Therefore, how to reduce the execution latency within the constraints of limited energy consumption is still an issue. In this paper, we focus on finding the optimal task offloading decision and resource allocation to decrease UE delay due to the constraint of energy consumption.

C. MAIN CONTRIBUTION

In this paper, we investigate the joint computation offloading and resource allocation optimization for IoT UEs in energy constrained MEC enabled HetNets. Our main contributions are summarized as follows:

(1) In the scenario of MEC-enabled HetNets, we first calculate the execution latency and energy consumption for each UE who chooses either local computing or edge computing. We also propose an appropriate interference management model for all UEs in the network in order to address the inter-user interference problem. Then, the weighted-sum execution latency of UEs is adopted as the performance metric, which can address the cost of execution latency at different nodes in MEC-enabled HetNets. The computation offloading decision and available resources management in the network are jointly optimized, including the CPU-cycle frequencies for each UE, the sub-channel and transmit power allocation for computation offloading, and the task offloading decision for each UE.

(2) Since the joint optimization problem we formulated is a mixed integer non-linear programming (MINLP) problem, the optimal solution can't be easily obtained. Thus,

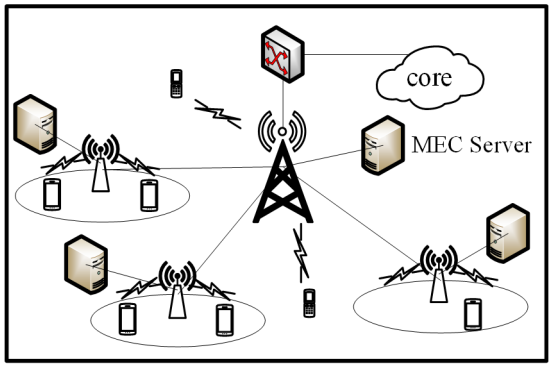

FIGURE 1. System model.

we decompose the problem into multiple sub-problems, including optimal local computing resource allocation, joint sub-channel and power allocation optimization, and the task offloading decision. First, we obtain the optimal close solution for the local CPU-cycle frequencies for each UE. For the sub-problem of joint sub-channel and power allocation optimization, we propose a low complexity algorithm using the CVX tool and the successive convex approximation (SCA) approach. Finally, combining the two sub-problems mentioned above, we propose the task offloading algorithm for each UE.

(3) We conduct a performance analysis of the proposed algorithm and we compare it with other related algorithms. The simulation results show that the proposed algorithm has better performance in execution latency. Moreover, the impact of the various parameters is determined, including the interference threshold, the sub-channel bandwidth, the weight coefficient, etc. Some interesting results are found, and the conclusion can offer valuable guidelines for real deployment.

D. ORGANIZATION OF THE PAPER

This paper is organized as follows. Section I introduces the research background, motivation, related works and the main work presented in this paper. Section II describes the system model and problem formulation. Section III demonstrates the proposed algorithm. Section IV presents the results of the simulation and the analysis. Section V presents the conclusion and suggestions for our future work.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we present the equations used to calculate execution latency and energy consumption for the local execution and the MEC server execution model, respectively. Then, we discuss the interference management model we proposed to eliminate the inter-user interference in the network. Finally, we formulate the joint computation offloading and resource allocation optimization problem for IoT UEs in MEC-enabled HetNets.

As shown in Fig.1, we consider a two-tier HetNets scenario that consists of the macro-BS (MBS) and small BSs (SBSs) in the network. The SBSs and UEs are randomly distributed within the coverage area of the MBS. Each BS is equipped with an MEC server for executing the tasks offloaded by the connected UEs. All the UEs can be connected to the core network through the MBS. The backhaul link between each SBS and the MBS can be regarded as an ideal link. For presentation, the set of BSs is denoted as $\mathcal{J} = \{0, 1, 2, \ldots, J\}$, where index 0 represents the MBS and index $1 - J$ represent the SBSs. The set of UEs is defined as $\{\mathcal{K}_j, j \in \mathcal{J}\}$, where $\mathcal{K}_j$ denotes the set of UEs associated with the BS j. The available system bandwidth W is divided into N sub-channels and the set of sub-channels is denoted as $\mathcal{N} = \{1, 2, \ldots, N\}$.

The computation tasks are generated by UEs and let $c_{i,j} = \left\{b_{i,j}, s_{i,j}, l_{i,j}, T_{i,j}^{\max}\right\}$ denotes the computation task generated by the UE i associated with the BS j. Here, $b_{i,j}$ and $s_{i,j}$ represent the number of computation tasks and the size of each task generated by UE i associated with the BS j. $l_{i,j}$ denotes the number of CPU cycles required to execute an one-bit task, which may vary with different applications. $T_{i,j}^{\max}$ is the maximum execution latency required by each computation task. There are two ways that each UE task can be executed: local execution and MEC server execution by the offloading process. The execution latency and energy consumption are different in these two options. The execution latency and energy consumption for the Local Execution and the MEC Server Execution Model are discussed below.

A. LOCAL EXECUTION MODEL

We assume the CPU frequency of each UE is fixed when computing, but it may vary over the UEs. Moreover, we ignore the effect of the stochastic task queue model on computation latency [22]. Let $f_{i,j}^{L}$ denote the local CPU frequency of the UE i associated with the BS j. Then the execution latency and energy consumption for local execution model can be calculated as shown in the Eq. (1) and Eq. (2), respectively.

$$t_{i,j}^{L} = \frac{b_{i,j}s_{i,j}l_{i,j}}{f_{i,j}^{L}}, \quad i \in \mathcal{K}_j,\ j \in \mathcal{J} \tag{1}$$

$$e_{i,j}^{L} = \kappa_{mob,i,j}\left(f_{i,j}^{L}\right)^2 b_{i,j}s_{i,j}l_{i,j}, \quad i \in \mathcal{K}_j,\ j \in \mathcal{J} \tag{2}$$

Here, $\kappa_{mob,i,j}$ is a coefficient related to the chip architecture of devices [23]. According to Eq. (1) and Eq. (2), the local CPU frequency $f_{i,j}^{L}$ may affect the execution latency and energy consumption simultaneously. The local CPU frequency of UEs can be adjusted using dynamic voltage and frequency scaling technology [24].

B. MEC SERVER EXECUTION MODEL

Apart from the local execution, the UEs can also choose to offload the computation task to MEC servers at the MBS or the SBSs. The computation offloading process mainly contains three phases: 1) uploading data to the associated MEC server through the allocated sub-channels, 2) computing the task on the MEC server, and 3) downloading the computation results from the MEC server to the UEs. The

execution latency for the proposed MEC Server Execution Model is shown in Eq. (3).

$$t_{i,j}^{C} = t_{i,j}^{UL} + t_{i,j}^{CP} + t_{i,j}^{DL} \approx t_{i,j}^{UL} + t_{i,j}^{CP} \tag{3}$$

Here, $t_{i,j}^{UL}$ denotes the uplink transmission latency for the UE i associated with the BS j. $t_{i,j}^{CP}$ is the computation latency on the MES server. $t_{i,j}^{DL}$ represents the downlink transmission delay from the MEC server to the UEs. Due to the small size of the computation result and the higher downlink transmission rate, the downlink transmission delay can be ignored [6], [25]. Next, we calculate the task uploading latency and computation latency on the MES server.

We assume that the sub-channel n is allocated for upload phase by the UE i associated with the BS j. According to *Shannon* theory, the available uplink transmission rate for the UE i associated with the BS j on the sub-channel n can be expressed as shown in the Eq. (4).

$$r_{i,j,n} = B\log_2\left(1 + \frac{p_{i,j,n}h_{i,j,n}^{j}}{I_{i,j,n} + B\sigma^2}\right) \tag{4}$$

Here, B denotes the bandwidth of the sub-channel n and it has $B = W/N$. $p_{i,j,n}$ is the transmit power for the UE i associated with the BS j on the sub-channel n. $h_{i,j,n}^{j'}$ represents the channel gain from the UE i associated with the BS j to the BS j' on the sub-channel n. σ^2 is the density of noise power. $I_{i,j,n}$ denotes the interference from the other UEs associated with the neighboring BSs, which is calculated in the interference management model.

Then, the uplink transmission rate and latency are given as shown in the Eq. (5) and Eq. (6), respectively.

$$r_{i,j} = \sum_{n\in\mathcal{N}} a_{i,j,n}r_{i,j,n} \tag{5}$$

$$t_{i,j}^{UL} = b_{i,j}s_{i,j}/r_{i,j} \tag{6}$$

Here, $a_{i,j,n} \in \{0, 1\}$, and $a_{i,j,n} = 1$ denotes the sub-channel n is allocated to the UE i associated with the BS j, otherwise, $a_{i,j,n} = 0$.

Let f^C denote the CPU frequency for the MEC server, which can be regarded as the fixed value for the duration of the computation phase for the tasks [26]. The computation latency on the MES server can be calculated as shown in the Eq. (7).

$$t_{i,j}^{CP} = b_{i,j}s_{i,j}l_{i,j}/f^{C}, \quad i \in \mathcal{K}_j,\ j \in \mathcal{J} \tag{7}$$

Moreover, the energy consumption of the UEs for the MEC Server Execution Model can be calculated as shown in Eq. (8).

$$e_{i,j}^{C} = \sum_{n=1}^{N} a_{i,j,n}p_{i,j,n}\frac{b_{i,j}s_{i,j}}{r_{i,j}} \tag{8}$$

Here, we assume the data transmission time over all the sub-channels is the same.

C. INTERFERENCE MANAGEMENT MODEL

The main reason for inter-user interference is that UEs upload the computation tasks to the MEC server on the same sub-channel. Due to the heterogeneity of HetNets, the interferences include cross-tier interference and co-tier interference. Appropriate sub-channel allocation and power control for UEs can effectively mitigate the interference [27].

We adopt the underlay mode for spectrum reuse to manage the cross-tier interference [28]. The mode allows the UEs associated with the MBS and the UEs associated with SBSs to share the sub-channels, but the interference to the UEs that is associated with the MBS generated by the UEs that is associated with the SBSs cannot exceed a threshold $I_{th,n}$, which is shown in Eq. (9).

$$\sum_{j\in\mathcal{J},j\neq 0}\sum_{i\in\mathcal{K}_j} a_{i,j,n}p_{i,j,n}h_{i,j,n}^{0} \le I_{th,n}, \quad n \in \mathcal{N} \tag{9}$$

Furthermore, the sub-channel allocation within the UEs of each cell and the SBSs are orthogonal, in order to avoid the interference among the UEs of each cell and the small cells, as shown in Eq. (10) and Eq. (11).

$$\sum_{j\in\mathcal{J},j\neq 0}\sum_{i\in\mathcal{K}_j} a_{i,j,n} \le 1, \quad n \in \mathcal{N} \tag{10}$$

$$\sum_{i\in\mathcal{K}_0} a_{i,j,n} \le 1, \quad n \in \mathcal{N} \tag{11}$$

Therefore, $I_{th,n}$ can be calculated as shown in the Eq. (12).

$$I_{i,j,n} = \begin{cases} \sum_{j'=1}^{J}\sum_{i'\in\mathcal{K}_{j'}} a_{i',j',n}p_{i',j',n}h_{i',j',n}^{0}, & if\ j = 0 \\ \sum_{i'\in\mathcal{K}_0} a_{i',0,n}p_{i',0,n}h_{i',0,n}^{j}, & if\ j \neq 0 \end{cases} \tag{12}$$

D. PROBLEM FORMULATION

Let $\lambda_{i,j}$ denote the computation offloading decision variable for the UE i associated with the BS j. $\lambda_{i,j} \in \{0, 1\}$. If the UE i decide to offload its computation task to the MEC server, $\lambda_{i,j} = 1$. Otherwise, $\lambda_{i,j} = 0$. Then, the execution latency and energy consumption for the computation offloading decision can be calculated as shown in Eq. (13) and Eq. (14), respectively.

$$t_{i,j}^{P} = \lambda_{i,j}t_{i,j}^{L} + \left(1 - \lambda_{i,j}\right)t_{i,j}^{C}, \quad i \in \mathcal{K}_j,\ j \in \mathcal{J} \tag{13}$$

$$e_{i,j}^{P} = \lambda_{i,j}e_{i,j}^{L} + \left(1 - \lambda_{i,j}\right)e_{i,j}^{C}, \quad i \in \mathcal{K}_j,\ j \in \mathcal{J} \tag{14}$$

In the paper, the computation offloading decision and the available radio and computational resources allocation are jointly optimized for IoT UEs in MEC-enabled HetNets, including the CPU-cycle frequencies for each UE, the sub-channel and transmit power allocation for computation offloading, and the computation offloading decision for each UE. The weighted-sum execution latency of the UEs was adopted as the performance metric, which can address the cost of execution latency at different nodes in MEC-enabled

HetNets. Then, the problem is formulated as shown in $P1$.

$$P1:$$
$$\min_{\left\{\lambda_{i,j},f^L_{i,j},a_{i,j,n},p_{i,j,n}\right\}} \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{K}_j}\omega_{i,j}t^P_{i,j} \tag{15}$$
$$s.t.\ C1: 0\le t^P_{i,j}\le T^{\max}_{i,j},\quad \forall i\in\mathcal{K}_j,\ j\in\mathcal{J} \tag{16}$$
$$C2: 0\le e^P_{i,j}\le E^{\max}_{i,j},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J} \tag{17}$$
$$C3: f^L_{i,j,\min}\le f^L_{i,j}\le f^L_{i,j,\max},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J} \tag{18}$$
$$C4: \sum_{j\in\mathcal{J},j\neq 0}\sum_{i\in\mathcal{K}_j} a_{i,j,n}p_{i,j,n}h^0_{i,j,n}\le I_{th,n},\quad n\in\mathcal{N} \tag{19}$$
$$C5: \sum_{n\in\mathcal{N}} a_{i,j,n}p_{i,j,n}\le p^{\max}_{i,j},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J} \tag{20}$$
$$C6: 0\le p_{i,j,n}\le p^{\max}_{i,j,n},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J},\ n\in\mathcal{N} \tag{21}$$
$$C7: \sum_{j\in\mathcal{J},j\neq 0}\sum_{i\in\mathcal{K}_j} a_{i,j,n}\le 1,\quad n\in\mathcal{N} \tag{22}$$
$$C8: \sum_{i\in\mathcal{K}_0} a_{i,j,n}\le 1,\quad n\in\mathcal{N} \tag{23}$$
$$C9: a_{i,j,n}\in\{0,1\},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J},\ n\in\mathcal{N} \tag{24}$$
$$C10: \lambda_{i,j}\in\{0,1\},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J} \tag{25}$$

Here, $\omega_{i,j}$ is the weight factor of the UE i associated with the BS j, which is introduced to represent variations in the different preference degree for different UEs. $C1$ denotes that the computation tasks needs to be executed within the required execution latency $T^{\max}_{i,j}$. $C2$ denotes that the energy consumption for the execution can't exceed the maximum value $E^{\max}_{i,j}$. $C3$ denotes a constraint that the local CPU frequency should be limited within a range of $\left[f^L_{i,j,\min},f^L_{i,j,\max}\right]$. $p^{\max}_{i,j}$ and $p^{\max}_{i,j,n}$ are the maximum transmit power of the UE i over all sub-channels and its power mask on sub-channel n, respectively. $C4$, $C7$ and $C8$ are the constraint of interference management model. $C9$ and $C10$ indicate the variables of sub-channel allocation and offloading decision variables are $0-1$ binary variables, respectively.

III. JOINT COMPUTATION OFFLOADING AND RESOURCE ALLOCATION ALGORITHM

In this section, we introduce the joint computation offloading and resource allocation algorithm for IoT UEs in MEC-enabled HetNets. Since the joint optimization problem we formulated is an (*MINLP*) problem, the optimal solution can't be easily obtained. Thus, we decompose the problem into multiple sub-problems, including optimal local computing resource allocation, joint sub-channel and power allocation optimization, and the task offloading decision. Next, we propose a corresponding solution for each sub-problem.

A. OPTIMAL LOCAL COMPUTING RESOURCE ALLOCATION

To minimize the weight-sum execution latency, the local CPU frequency can be scheduled dynamically. Considering the constraints of $C1-C3$ in $P1$, $P2$ can be formulated as follows.

$$P2: \min_{\left\{f^L_{i,j}\right\}} \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{K}_j}\omega_{i,j}t^L_{i,j}\left(\mathbf{f}^L\right) \tag{26}$$
$$s.t.\ C1: 0\le t^L_{i,j}\le T^{\max}_{i,j},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J} \tag{27}$$
$$C2: 0\le e^L_{i,j}\le E^{\max}_{i,j},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J} \tag{28}$$
$$C3: f^L_{i,j,\min}\le f^L_{i,j}\le f^L_{i,j,\max},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J} \tag{29}$$

Here, $\mathbf{f}^L=\left\{f^L_{i,j},i\in\mathcal{K}_j,j\in\mathcal{J}\right\}$.

According to the constraints of $C1$ and $C2$, the variable $f^L_{i,j}$ needs to satisfy Eq. (30).

$$f^L_{i,j}\ge\frac{b_{i,j}s_{i,j}l_{i,j}}{T^{\max}_{i,j}}=f^{L,down}_{i,j}$$
$$0\le f^L_{i,j}\le\sqrt[3]{\frac{E^{\max}_{i,j}}{\kappa_{mob,i,j}b_{i,j}s_{i,j}l_{i,j}}}=f^{L,up}_{i,j} \tag{30}$$

Combined with the constraint of $C3$, we have $f^{L,down}_{i,j}\le f^{L,up}_{i,j}$ and $f^{L,down}_{i,j}\le f^L_{i,j,\max}$, which ensures $P2$ has the nonempty feasible domain. It is noted that the function $t^L_{i,j}\left(\mathbf{f}^L\right)$ is monotonically decreasing on the domain. Hence, we obtain the optimal close solution for the local CPU-cycle frequencies for each UE, which is shown in Eq. (31).

$$f^{L*}_{i,j}=\min\left(f^{L,up}_{i,j},f^L_{i,j,\max}\right) \tag{31}$$

B. ITERATIVE JOINT SUB-CHANNEL AND POWER ALLOCATION ALGORITHM

To solve $P1$, we propose a joint uplink transmission power and sub-channel allocation optimization algorithm for each UE.

According to the MEC Server Execution Model, the execution latency of the computation task can be expressed as $t^C_{i,j}=t^{UL}_{i,j}+t^{CP}_{i,j}$. By combining this with the Eq. (7), we can reformulate the problem $P1$ into the problem $P3$, because $t^{CP}_{i,j}$ can be regarded as the fixed value and it has nothing to do with the uplink transmission power and the sub-channel allocation. And the problem $P3$ is shown as follows.

$$P3: \min_{\{a_{i,j,n},p_{i,j,n}\}} \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{K}_j}\omega_{i,j}t^{UL}_{i,j} \tag{32}$$
$$s.t.\ C1: 0\le t^{UL}_{i,j}\le T^{\max}_{i,j}-t^{CP}_{i,j},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J} \tag{33}$$
$$C2: 0\le e^C_{i,j}\le E^{\max}_{i,j},\quad i\in\mathcal{K}_j,\ j\in\mathcal{J} \tag{34}$$
$$C4\sim C9 \tag{35}$$

According to Eq. (8), we can obtain that the uplink transmission latency is inversely proportional to the available rate of uplink transmission. The objective function in $P3$ can be transformed into the objective

Algorithm 1 Iterative Joint Sub-Channel and Power Allocation Algorithm

1: Initialize $\mathbf{p}(0)$ and $l = 1$.
2: **repeat**
3: given value of $\mathbf{p}(l-1)$, calculate optimal sub-channel allocation $\mathbf{a}(l)$ by solving the problem $P(3-2)$;
4: with the fixed value of $\mathbf{a}(l)$, calculate the optimal transmit power $\mathbf{p}(l)$ by solving the problem $P(3-3)$;
5: Update $l = l + 1$ and ξ^l;
6: **until** $\xi^l \leq \sigma_1$
7: Return the optimal solution by $\mathbf{p}^* = \mathbf{p}(l)$ and $\mathbf{a}^* = \mathbf{a}(l)$

in $P(3-1)$, with the definition of $\hat{\omega}_{i,j} = 1/b_{i,j}s_{i,j}\omega_{i,j}$. We also make some changes to the constraints of $C1$, $C2$ and $C5$, with the definition of $r^u_{i,j} = b_{i,j}s_{i,j}/\left(T^{\max}_{i,j} - t^{CP}_{i,j}\right)$ and $\pi^{\max}_{i,j} = \min\left(p^{\max}_{i,j}, E^{\max}_{i,j}r_{i,j}/b_{i,j}s_{i,j}\right)$. Furthermore, the constraint $C4$, $C6-C9$ in $P3$ is the same as the constraint $C4$, $C6-C9$ in $P(3-1)$. To sum up, for the optimization variables $\{a_{i,j,n}, p_{i,j,n}\}$, the problem $P(3-1)$ is equal to $P3$.

$$P(3-1): \max_{\{a_{i,j,n},p_{i,j,n}\}} \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{K}_j} \hat{\omega}_{i,j}r_{i,j} \tag{36}$$

$$s.t.\ C1: r_{i,j} \geq r^u_{i,j}, \quad i \in \mathcal{K}_j,\ j \in \mathcal{J} \tag{37}$$

$$C2: \sum_{n\in\mathcal{N}} a_{i,j,n}p_{i,j,n} \leq \pi^{\max}_{i,j}, \quad i \in \mathcal{K}_j,\ j \in \mathcal{J} \tag{38}$$

$$C4 \sim C9 \tag{39}$$

To solve the problem in $P(3-1)$, the iteration optimization algorithm for sub-channel and power allocation is adopted, which is shown in *Algorithm* 1. In *Algorithm* 1, $\mathbf{p}(l)$ and $\mathbf{a}(l)$ denote the set of variables related to transmit power and sub-channel allocation for UEs in the lth iteration, respectively. Specifically, each iteration can be divided into two steps. In step 1, the optimal sub-channel allocation of iteration l is derived from the given value of $\mathbf{p}(l-1)$. Then, with the fixed value of $\mathbf{a}(l)$, the optimal transmit power is obtained. The iterative algorithm is repeated until it converges, then we can obtain the optimal solution of the sub-channel and power allocation for $P(3-1)$. The iterative algorithm will converge until $\xi^l \leq \sigma_1$ and the ξ^l can be defined as the Eq. (40) or Eq. (41).

$$\xi^l = \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{K}_j} \hat{\omega}_{i,j}r_{i,j}(l) - \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{K}_j} \hat{\omega}_{i,j}r_{i,j}(l-1) \tag{40}$$

$$\xi^l = \|\mathbf{p}(l) - \mathbf{p}(l-1)\| \tag{41}$$

As shown in *Algorithm* 1, we need to solve $P(3-2)$ and $P(3-3)$ to obtain the optimal sub-channel and power allocation, respectively. The problem $P(3-2)$ is shown as follows.

$$P(3-2): \max_{\{\mathbf{a}(l)\}} \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{K}_j} \hat{\omega}_{i,j}r_{i,j}(\mathbf{a}(l), \mathbf{p}(l-1)) \tag{42}$$

$$s.t.\ C1: \sum_{n\in\mathcal{N}} a_{i,j,n}(l)\, r_{i,j,n}(\mathbf{p}(l-1)) \geq r^u_{i,j} \tag{43}$$

$$C2: \sum_{n\in\mathcal{N}} a_{i,j,n}(l)\, p_{i,j,n}(l-1) \leq \pi^{\max}_{i,j} \tag{44}$$

$$C4: \sum_{j\in\mathcal{J},j\neq 0}\sum_{i\in\mathcal{K}_j} a_{i,j,n}(l)\, p_{i,j,n}(l-1)\, h^0_{i,j,n} \leq I_{th,n} \tag{45}$$

$$C7: \sum_{j\in\mathcal{J},j\neq 0}\sum_{i\in\mathcal{K}_j} a_{i,j,n}(l) \leq 1 \tag{46}$$

$$C8: \sum_{i\in\mathcal{K}_0} a_{i,j,n}(l) \leq 1 \tag{47}$$

$$C9: a_{i,j,n}(l) \in \{0,1\} \tag{48}$$

Here, $r_{i,j}(\mathbf{a}(l), \mathbf{p}(l-1))$ and $r_{i,j,n}(\mathbf{p}(l-1))$ is shown as follows.

$$r_{i,j}(\mathbf{a}(l), \mathbf{p}(l-1)) = \sum_{n\in\mathcal{N}} a_{i,j,n}r_{i,j,n}(\mathbf{p}(l-1)) \tag{49}$$

$$r_{i,j,n}(\mathbf{p}(l-1)) = \log_2\left(1 + h^j_{i,j,n}p_{i,j,n}(l-1)/I_{i,j,n}(l-1) + \sigma^2\right) \tag{50}$$

Due to the existence of the integer variable of $\mathbf{a}(l)$, $P(3-2)$ is a non-convex integer programming problem, it is difficult to obtain a closed expression for the optimal solution. Moreover, this problem will cause high computational complexity when using an exhaustive search method. Realistically, we need to propose a low computational complexity algorithm to solve the problem $P(3-2)$. Toward that end, we convert the discrete variables $a_{i,j,n}(l) \in \{0,1\}$ into continuous variables $a_{i,j,n}(l) \in [0,1]$ approximately, which refers to the proportion of time allocated to the UE on each sub-channel, and then obtain the suboptimal solution for $P(3-2)$. Thus, $P(3-2)$ is transformed into a convex problem and it can be solved by utilizing available online optimization software, e.g., the CVX tool [29]. With the fixed value of $\mathbf{a}(l)$, $P(3-1)$ can be transformed into $P(3-3)$, which can be shown as follows. Due to the rate function $r_{i,j,n}(\mathbf{p}(l))$ is a highly non-concave function, $P(3-3)$ is obviously not a convex problem. In order to solve this problem, we use the SCA approach [30] to propose the power allocation algorithm with *logarithmic approximation*.

$$P(3-3): \max_{\{\mathbf{p}(l)\}} \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{K}_j} \hat{\omega}_{i,j}r_{i,j}(\mathbf{a}(l), \mathbf{p}(l)) \tag{51}$$

$$s.t.\ C1: \sum_{n\in\mathcal{N}} a_{i,j,n}(l)\, r_{i,j,n}(\mathbf{p}(l)) \geq r^u_{i,j} \tag{52}$$

$$C2: \sum_{n\in\mathcal{N}} a_{i,j,n}(l)\, p_{i,j,n}(l) \leq \pi^{\max}_{i,j} \tag{53}$$

$$C4: \sum_{j\in\mathcal{J},j\neq 0}\sum_{i\in\mathcal{K}_j} a_{i,j,n}(l)\, p_{i,j,n}(l)\, h^0_{i,j,n} \leq I_{th,n} \tag{54}$$

$$C6: 0 \leq p_{i,j,n}(l) \leq p^{\max}_{i,j,n} \tag{55}$$

Instead of directly dealing with the highly non-concave rate function, we apply the logarithmic approximation method to convert the rate function into $\hat{r}_{i,j,n}\left(\mathbf{p}\left(l_p\right)\right)$, which is shown in Eq. (56).

$$\hat{r}_{i,j,n}\left(\mathbf{p}\left(l_p\right)\right) = \alpha_{i,j,n}\left(l_p\right) + \beta_{i,j,n}\left(l_p\right)\log_2\left(\gamma_{i,j,n}\left(\mathbf{p}\left(l_p\right)\right)\right) \tag{56}$$

Here, $\alpha_{i,j,n}\left(l_p\right)$ and $\beta_{i,j,n}\left(l_p\right)$ are the parameters for approximation, which can be calculated as seen in Eq. (57) and Eq. (58), respectively. $\hat{r}_{i,j,n}\left(\mathbf{p}\left(l_p\right)\right)$ denotes the low bound of the original rate function $r_{i,j,n}\left(\mathbf{p}\left(l\right)\right)$.

$$\begin{aligned}\alpha_{i,j,n}\left(l_p\right) &= \log_2\left(1+\gamma_{i,j,n}\left(\mathbf{p}\left(l_p-1\right)\right)\right)\\ &\quad -\beta_{i,j,n}\left(l_p\right)\log_2\left(\gamma_{i,j,n}\left(\mathbf{p}\left(l_p-1\right)\right)\right) \qquad (57)\\ \beta_{i,j,n}\left(l_p\right) &= \gamma_{i,j,n}\left(\mathbf{p}\left(l_p-1\right)\right)\big/\left(1+\gamma_{i,j,n}\left(\mathbf{p}\left(l_p-1\right)\right)\right) \qquad (58)\end{aligned}$$

Due to the existence of logarithmic function $\log_2\left(\gamma_{i,j,n}\left(\mathbf{p}\left(l_p\right)\right)\right)$, the function $\hat{r}_{i,j,n}\left(\mathbf{p}\left(l_p\right)\right)$ is still a non-concave function. Let $\hat{\mathbf{p}} = log_2\mathbf{p}$, then, the logarithmic function can be converted into a $log-sum-exp$ function and the problem $P(3-3)$ can be reformulated into the problem $P(3-4)$, which can be easily proved as a convex problem [31]. The problem $P(3-4)$ is shown as follows.

$$P(3-4): \max_{\{\hat{\mathbf{p}}(l_p)\}} \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{K}_j} \hat{\omega}_{i,j}\hat{r}_{i,j,n}\left(e^{\hat{\mathbf{p}}(l_p)}, \boldsymbol{\alpha}(l_p), \boldsymbol{\beta}\left(l_p\right)\right) \tag{59}$$

$$s.t.\ C1: \sum_{n\in\mathcal{N}} a_{i,j,n}\left(l\right)\hat{r}_{i,j,n}\left(e^{\hat{\mathbf{p}}(l_p)}, \boldsymbol{\alpha}\left(l_p\right), \boldsymbol{\beta}\left(l_p\right)\right) \geq r^u_{i,j} \tag{60}$$

$$C2: \sum_{n\in\mathcal{N}} a_{i,j,n}\left(l\right)e^{\hat{p}_{i,j,n}(l_p)} \leq \pi^{\max}_{i,j} \tag{61}$$

$$C4: \sum_{j\in\mathcal{J},j\neq 0}\sum_{i\in\mathcal{K}_j} a_{i,j,n}\left(l\right)e^{\hat{p}_{i,j,n}(l_p)}h^0_{i,j,n} \leq I_{th,n} \tag{62}$$

$$C6: 0 \leq e^{\hat{p}_{i,j,n}(l_p)} \leq p^{\max}_{i,j,n} \tag{63}$$

Here, $\boldsymbol{\alpha}\left(l_p\right) = \left\{\alpha_{i,j,n}\left(l_p\right)\middle|\, i\in\mathcal{K}_j, j\in\mathcal{J}, n\in\mathcal{N}\right\}$ and $\boldsymbol{\beta}\left(l_p\right) = \left\{\beta_{i,j,n}\left(l_p\right)\middle|\, i\in\mathcal{K}_j, j\in\mathcal{J}, n\in\mathcal{N}\right\}$. In HetNets without a central processing unit (e.g., when the MBS and SBSs belong to different service providers), it is more desirable for UEs to distribute and control the transmit power. Utilizing Lagrangian duality, we propose the SCA-based power allocation algorithm based on the *logarithmic approximation method*. The Lagrangian duality of the problem is defined as Eq. (64).

$$\begin{aligned}&\mathcal{L}\left(\hat{\mathbf{p}}\left(l_p\right), \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}\right)\\ &= \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{K}_j}\hat{\omega}_{i,j}\hat{r}_{i,j,n}\left(e^{\hat{\mathbf{p}}(l_p)}, \boldsymbol{\alpha}\left(l_p\right), \boldsymbol{\beta}\left(l_p\right)\right)\\ &\quad + \lambda_{i,j}\left(l_p\right)\left[\sum_{n\in\mathcal{N}} a_{i,j,n}\left(l\right)\hat{r}_{i,j,n}\left(e^{\hat{\mathbf{p}}(l_p)}, \boldsymbol{\alpha}(l_p), \boldsymbol{\beta}\left(l_p\right)\right) - r^u_{i,j}\right]\\ &\quad - \mu_{i,j}\left(l_p\right)\left[\sum_{n\in\mathcal{N}} a_{i,j,n}\left(l\right)e^{\hat{p}_{i,j,n}(l_p)} - \pi^{\max}_{i,j}\right]\\ &\quad - \nu_n\left(l_p\right)\left[\sum_{j\in\mathcal{J},j\neq 0}\sum_{i\in\mathcal{K}_j} a_{i,j,n}\left(l\right)e^{\hat{p}_{i,j,n}(l_p)}h^0_{i,j,n} - I_{th,n}\right]\end{aligned} \tag{64}$$

Here, $\lambda_{i,j}\left(l_p\right)$, $\mu_{i,j}\left(l_p\right)$ and $\nu_n\left(l_p\right)$ are the Lagrangian multipliers associated with $C1$, $C2$ and $C4$ of the problem $P(3-4)$, respectively. The dual problem of $P(3-4)$ can be shown as follows.

$$D\left(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}\right) = \max_{\{\hat{\mathbf{p}}(l_p)\}} \mathcal{L}\left(\hat{\mathbf{p}}\left(l_p\right), \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}\right) \tag{65}$$

$$\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu} \geq 0, C6 \tag{66}$$

After solving the stationary condition of the Eq. (64) by $\partial\mathcal{L}\left(\hat{\mathbf{p}}\left(l_p\right), \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}\right)\big/\partial\hat{\mathbf{p}}\left(l_p\right) = 0$ and transforming the result to the $\mathbf{p}$-space, we can obtain the Eq. (67), as shown at the bottom of the next page.

Then, the solution of the dual problem $\min_{\{\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\nu}\}} D\left(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}\right)$ can be determined by a *subgradient* method as shown in Eq. (68)-(70).

$$\lambda_{i,j}\left(l_p, t+1\right) = \left[\lambda_{i,j}\left(l_p, t\right) + \varepsilon_\lambda\left(r^u_{i,j} - C_\lambda\left(l_p, t\right)\right)\right]^+ \tag{68}$$

$$\mu_{i,j}\left(l_p, t+1\right) = \left[\mu_{i,j}\left(l_p, t\right) + \varepsilon_\mu\left(C_\mu\left(l_p, t\right) - \pi^{\max}_{i,j}\right)\right]^+ \tag{69}$$

$$\nu_n\left(l_p, t+1\right) = \left[\nu_n\left(l_p, t\right) + \varepsilon_\mu\left(C_\nu\left(l_p, t\right) - I_{th,n}\right)\right]^+ \tag{70}$$

Here, $[x]^+$ represents $\max\left(x, 0\right)$, and $C_\lambda\left(l_p, t\right)$, $C_\mu\left(l_p, t\right)$ and $C_\nu\left(l_p, t\right)$ can be represented as Eq. (71)-Eq. (73).

$$C_\lambda\left(l_p, t\right) = \sum_{n\in\mathcal{N}} a_{i,j,n}\left(l\right)\hat{r}_{i,j,n}\left(\mathbf{p}\left(l_p, t\right), \boldsymbol{\alpha}(l_p), \boldsymbol{\beta}\left(l_p\right)\right) \tag{71}$$

$$C_\mu\left(l_p, t\right) = \sum_{n\in\mathcal{N}} a_{i,j,n}\left(l\right)p_{i,j,n}\left(l_p, t\right) \tag{72}$$

$$C_\nu\left(l_p, t\right) = \sum_{j\in\mathcal{J},j\neq 0}\sum_{i\in\mathcal{K}_j} a_{i,j,n}\left(l\right)p_{i,j,n}\left(l_p, t\right)h^0_{i,j,n}. \tag{73}$$

As mentioned above, the *SCA*-based power allocation algorithm based on the *logarithmic approximation* method is given as shown in *Algorithm* 2.

C. DISTRIBUTED COMPUTATION OFFLOADING ALGORITHM

In MEC enabled HetNets without a central processing unit (*e.g.*, when MBS and SBSs belong to different service providers), it will be more desirable for UEs to distributedly make the computation offloading decision. Moreover, we can obtain the correspondingly optimal execution latency for the local execution and MEC server execution model by Eq. (31) and *Algorithm* 1, which can be denoted by $t^{L*}_{i,j}$ and $t^{C*}_{i,j}$, respectively. Then, each UE can choose the model with less execution latency to complete the tasks. The corresponding computation decision variables and the weight-sum execution

Algorithm 2 SCA-Based Power Allocation Algorithm Based on *Logarithmic Approximation*

1: Initialize $\boldsymbol{\alpha}(0)$, $\boldsymbol{\beta}(0)$.
2: **repeat**to solve $P(3-2)$
3: **repeat**to solve $P(3-3)$
4: According to the Eq. (67), calculate $p_{i,j,n}(l_p, t)$;
5: According to the Eq. (68)-(70), update $\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}$;
6: **until** $\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}$ converges;
7: Let $p_{i,j,n}(l_p) = p_{i,j,n}(l_p, t)$;
8: According to the Eq. (57) and Eq. (58), calculate $\alpha_{i,j,n}(l_p+1)$ and $\beta_{i,j,n}(l_p+1)$;
9: Update $l_p = l_p + 1$;
10: **until** $\mathbf{p}$ converges;

Algorithm 3 Joint Computation Offloading and Resource Allocation Algorithm

Input: $b_{i,j}$, $s_{i,j}$, and $h^{j'}_{i,j,n}$.
Output: $\lambda^*_{i,j}$, $\mathbf{f}^{L*}$, $\mathbf{p}^*$ and $\mathbf{a}^*$.
1: According to Eq. (31) and Eq. (1), calculate the optimal local CPU frequency $\mathbf{f}^{L*}$ and the correspond execution latency $t^{L*}_{i,j}$, respectively
2: According to *Algorithm* 1 and Eq. (3), calculate the solution of the sub-problem P3 ($\mathbf{p}^*$ and $\mathbf{a}^*$) and the corresponding execution latency $t^{C*}_{i,j}$, respectively;
3: According to Eq. (74) and Eq. (75), calculate the computation offloading decision and the corresponding execution latency $\lambda^*_{i,j}$, respectively;

latency for the network can be calculated as shown in Eq. (74) and Eq. (75), respectively.

$$\lambda^*_{i,j} = \begin{cases} 1, & t^{L*}_{i,j} < t^{C*}_{i,j} \\ 0, & t^{L*}_{i,j} \geq t^{C*}_{i,j} \end{cases} \tag{74}$$

$$\begin{aligned} T^* &= \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{K}_j} \omega_{i,j} t^{P*}_{i,j} \\ &= \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{K}_j} \omega_{i,j} \left\{ \lambda^*_{i,j} t^{L*}_{i,j} + \left(1-\lambda^*_{i,j}\right) t^{C*}_{i,j} \right\} \end{aligned} \tag{75}$$

As mentioned above, the joint computation offloading and resource allocation algorithm can be given as shown in *Algorithm* 3.

D. DISCUSSIONS

1) ANALYSIS OF OPTIMALITY

As shown in *Algorithm* 3, we propose a low complexity algorithm to solve the problem of computation offloading in energy-constrained MEC-enabled HetNets, which aims to minimize the weighted-sum delay over the users.

Proposition 1: For a feasible problem of $P1$, *Algorithm* 3 will obtain a local optimum for $P1$.

Proof: Algorithm 3 is primarily composed of three main procedures: optimal local computing resource allocation, joint optimization for sub-channel and power allocation (as shown in Algorithm 1), and computation offloading decision making. For the problem $P1$, if $t^{L*}_{i,j} < t^{C*}_{i,j}$, the optimal offloading decision should be determined as $\lambda^*_{i,j} = 1$. Otherwise, $\lambda^*_{i,j} = 0$. According to Eq. (1) and Eq. (3), $t^{L*}_{i,j}$ and $t^{C*}_{i,j}$ can be obtained by the optimal CPU frequency and the wireless resource allocation, respectively. The optimal CPU frequency for each user can be calculated by the close expression of Eq. (31). For the problem of joint sub-channel and power allocation optimization, the *Algorithm* 1 will converge to give a local maximum. the proof is shown as follows.

$$\begin{aligned} U(\mathbf{a}(l), \mathbf{p}(l)) &= \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{K}_j} \hat{\omega}_{i,j} r_{i,j}(\mathbf{a}(l), \mathbf{p}(l)) \\ &= \max_{\mathbf{p}} U(\mathbf{a}(l), \mathbf{p}) \\ &\geq U(\mathbf{a}(l), \mathbf{p}(l-1)) \\ &= \max_{\mathbf{a}} U(\mathbf{a}(l), \mathbf{p}(l-1)) \\ &\geq U(\mathbf{a}(l-1), \mathbf{p}(l-1)) \end{aligned} \tag{76}$$

This means that *Algorithm* 1 gives a non-decreasing objective function as the iterations continue. When the sequence of iterations converges, the related solution is feasible and a local maximum is obtained. Nevertheless, it should be noted that the algorithm often empirically achieves the globally optimal solution. Finally, the sub-optimal computation decision can be obtained.

2) COMPUTATIONAL COMPLEXITY

In this section, we discuss the computational complexity of the proposed joint computation offloading and resource allocation algorithm presented in this paper. According to *Algorithm* 3, the proposed algorithm can be divided into three parts, including the optimization of local CPU frequency, joint user sub-channel and power allocation, and the computation offloading decision. To optimize the local

$$p_{i,j,n}(l_p, t) = \begin{cases} \left[\sqrt{\dfrac{\left(\hat{\omega}_{i,j} + \lambda_{i,j}(l_p)\, a_{i,j,n}(l)\right)\beta}{\left(\mu_{i,j}(l_p)\, a_{i,j,n}(l)\right) In2}} \right]_0^{p^{\max}_{i,j,n}} \\ \left[\sqrt{\dfrac{\left(\hat{\omega}_{i,j} + \lambda_{i,j}(l_p)\, a_{i,j,n}(l)\right)\beta}{\left(\mu_{i,j}(l_p)\, a_{i,j,n}(l)\right) In2 + \nu_n(l_p)\, a_{i,j,n}(l)\, h^0_{i,j,n}}} \right]_0^{p^{\max}_{i,j,n}} \end{cases} \tag{67}$$

CPU frequency, we need to traverse through all the UEs to obtain the optimal local CPU frequency; the computational complexity of this procedure is $O\left(|\mathcal{J}|\left|\mathcal{K}_j\right|\right)$, where $|x|$ denotes the cardinality of the set of x. The joint sub-channel and power allocation issue can be addressed using CVX to solve the sub-channel allocation problem and using SCA to solve the power control problem. The computational complexity of the sub-channel allocation using CVX is approximately $O\left(\log\left(|\mathcal{J}|\left|\mathcal{K}_j\right|N\right)\right)$ [31]. The computational complexity using *SCA* to solve the power control problem is $O\left(|\mathcal{J}|\left|\mathcal{K}_j\right|N\right)\times O\left(2|\mathcal{J}|\left|\mathcal{K}_j\right|+N\right)$. Here, the power allocation is derived from Eq. (67), leading to the computational complexity of $O\left(|\mathcal{J}|\left|\mathcal{K}_j\right|N\right)$, and $O\left(2|\mathcal{J}|\left|\mathcal{K}_j\right|+N\right)$ is the computational complexity of updating dual variables according to Eqs. (68-70). Ultimately, we need to traverse through each UE's task offloading decision according to Eq. (74); the corresponding computational complexity is $O\left(|\mathcal{J}|\left|\mathcal{K}_j\right|\right)$.

3) IMPLEMENTATION AND SCALABILITY

Algorithm 3 consists of three key steps: (i) compute the local CPU frequency via Eq. (15), (ii) solve a joint optimization problem for the subchannel and power allocation through Algorithm 1, and (iii) make the computation offloading decision. To implement Algorithm 1, channel state information (CSI) is most likely needed for the system. The MBS and SBSs measure the CSI between their users and the interference CSI of the other users to their receivers on the uplink. Then, the SBSs send all the obtained CSI to the MBS. Moreover, a central processing unit in the MBS is most likely needed to collect all the network information and perform the proposed Algorithm 1 to obtain the $t_{i,j}^{C*}$. Then, each processing unit in an SBS can compute the optimal local CPU frequency and obtain the $t_{i,j}^{L*}$. With $t_{i,j}^{L*}$ and $t_{i,j}^{C*}$, each processing unit in the system can make the corresponding computation offloading decision for each task.

Note that multiple-antenna devices are a new trend for meeting the significantly higher traffic demands in the beyond 5G era [32]. The system model proposed in this paper considers single-antenna devices. However, the system model can be extended to the case of multiple-antenna devices with some modifications. The system model for the case of multiple-antenna devices is similar to the one formulated for single-antenna devices; however, in the system model for multiple-antenna devices there are some differences in the channel gains and uplink transmission rate. Therefore, the major modifications include replacing Eq. (4), presented in this paper, with Eq. (1), presented in [32]. Finally, the corresponding computation offloading decision and resource allocation can be obtained using the proposed *Algorithm* 3.

IV. SIMULATION RESULTS

In this paper, we consider a coverage area of $1km \times 1km$ for MEC-enabled HetNets, which includes an MBS and several SBSs. The MBS is located at the origin while the SBSs are randomly distributed in the area. UEs are randomly distributed within the coverage area of the associated BS. The number of computation tasks generated by each UE follows a *Poisson distribution* with a mean of 400. The data size of each computation task is $1KB$ and each computation task requires 10^9 CPU cycles to complete. The local CPU frequency of each UE ranges from $0.2GHz$ to 2 GHz and the CPU frequency of each MEC server is set as 4 GHz. The channel gain from each UE to each BS refers to *Model A.2.1.1.2-3 for the outdoor RRH or hotspot area model 1* [33] in the *3GPP* specifications. Other required parameters are shown in *Table* 1.

TABLE 1. Simulation parameters.

Simulation parameters	Default value
Radius of each SBS	$50m$
Number of SBSs	5
Number of macro-UEs	5
Number of small UEs	2
Number of sub-channels N	10
Sub-channel bandwidth B	$100KB$
Maximum transmit power $p_{i,j}^{\max}$	$23dBm$
Interference threshold $I_{th,n}$	$0dBm$
Density of noise power σ^2	$-174dBm/Hz$
Coefficient $\kappa_{mob,i.j}$	10^{-26}
Maximum energy consumption $E_{i,j}^{\max}$	$50\times10^{-3}J$
Maximum execution latency $T_{i,j}^{\max}$	$10s$
Weight of UEs $\omega_{i,j}$	1

The numerical experiments are performed using the Monte Carlo method. We compared the proposed algorithm with five baseline algorithms. The term, ''Optimal offloading'', denotes the proposed joint computation offloading and resource allocation algorithm in this paper. The terms, ''All local'' and ''All MEC'' represent all the computation tasks executed locally and by the MEC servers, respectively. The term, ''Exhaustive search'', means the computation offloading and resource allocation are optimized using the exhaustive search algorithm. The optimal solution is obtained by an exhaustive search algorithm; it adopts the enumerator to generate all feasible solutions, and then it selects the optimal one, which minimizes the weighted-sum delay in $P1$. ''RC'' and ''RP'' indicate a random sub-channel allocation strategy and random power allocation, respectively. All the simulations are performed on a desktop computer with an Intel Core i7-8700U 3.2 GHz CPU and 24 GB memory.

Fig. 2 shows the convergence of *Algorithm* 1 and *Algorithm* 2, respectively. Here, $|\Delta\mathbf{p}(l)|=\sum_{i,j,n}\left(p_{i,j,n}(l)-p_{i,j,n}(l-1)\right)$ and $\left|\Delta\mathbf{p}\left(l_p\right)\right|=\sum_{i,j,n}\left(p_{i,j,n}\left(l_p\right)-p_{i,j,n}\left(l_p-1\right)\right)$. It can be seen from Fig.2 that both *Algorithm* 1 and *Algorithm* 2 can converge with limited iterations.

Fig. 3 shows the performance of the weighted-sum execution latency and compares the proposed algorithm and other algorithms (including All MEC, All local, RP, RC, and Exhaustive search) as the number of small cells increases. From the information presented in Fig. 3, we can conclude

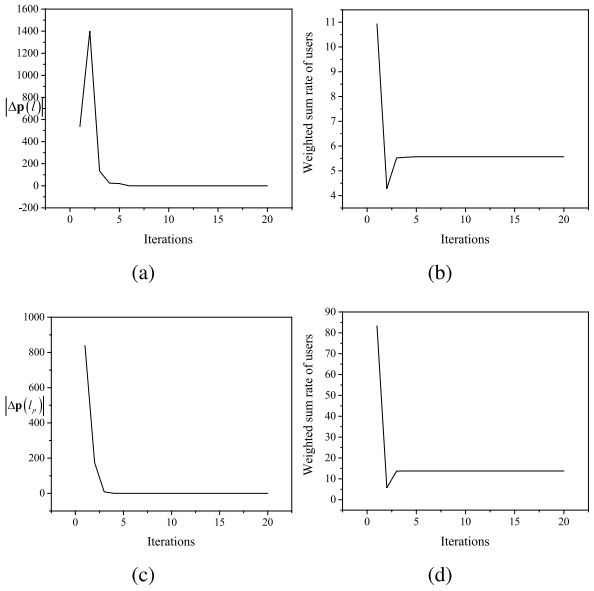

FIGURE 2. The convergence of proposed *Algorithm* 1 and *Algorithm* 2.

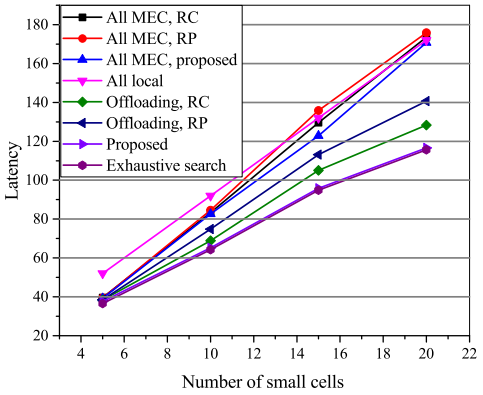

FIGURE 3. The performance of the weighted-sum execution latency under different algorithms.

that the weighted-sum execution latency increases linearly as the number of small cells increases. This is because the number of UEs increases linearly in the network, and the fixed sub-channels resource limits the UEs' available uplink transmit rate, which leads to a linear increase in execution latency. As seen in Fig. 3, we compare the performance of the proposed algorithm with the exhaustive search algorithm. The results demonstrate that the execution latency performance of the proposed algorithm is similar to that of the exhaustive search algorithm. It is important to note that we can only prove that the solution obtained by the proposed algorithm is feasible and has a local maximum. Nevertheless, it should be mentioned that the algorithm often empirically achieves the globally optimal solution, as is seen in Fig. 3. The exhaustive search algorithm has a computation complexity of $O\left(2^{|\mathcal{J}||\mathcal{K}_j|N}\right)$. As is shown in Fig. 2, the proposed algorithm has a faster convergence speed than the exhaustive search algorithm, and the computation complexity of each iteration

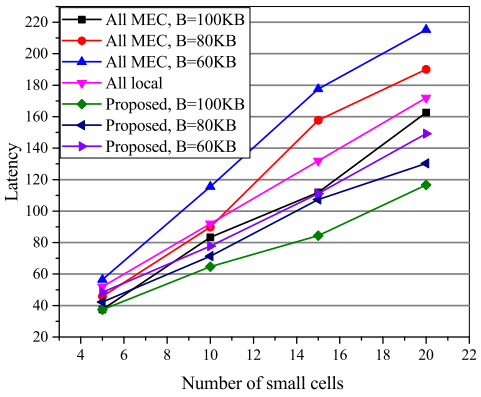

FIGURE 4. The effects of different sub-channel bandwidths on network latency.

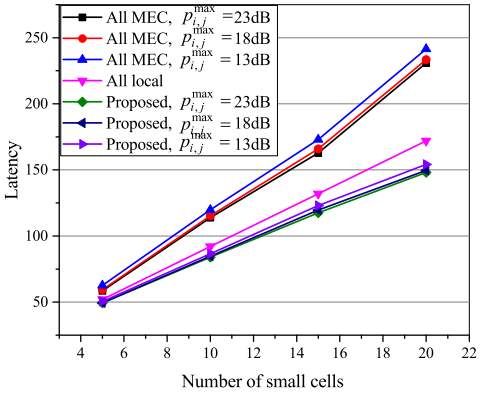

FIGURE 5. The effects of the maximum transmit power of different UEs on network latency.

process can be approximated to $O\left(|\mathcal{J}|\left|\mathcal{K}_j\right|N\right)$. The average execution time of the proposed algorithm and the exhaustive search algorithm is about 88.24s and 2.39e-2s, respectively. As the number of UEs and sub-channels increase, the proposed algorithm becomes more efficient than the exhaustive search algorithm. Moreover, the proposed algorithm has a better execution latency performance than the All MEC, All local, RC, and RP algorithms. It can be concluded that appropriate computation offloading decision and sub-channel and power allocation play an important role in the optimization problem of the weighted-sum execution latency.

Fig. 4 and Fig. 5 show the effects of different sub-channel bandwidths and maximum transmit power of UEs on the network latency obtained by the All MEC algorithm, the All local algorithm, and the proposed algorithm. It can be concluded that the sub-channel bandwidths and the maximum transmit power of UEs only have a significant impact on the weighted-sum execution latency obtained by the All MEC algorithm and the proposed algorithm. As the bandwidth of sub-channels increases, the weighted-sum execution latency obtained by these two algorithms decreases significantly. This is because the UEs' uplink transmit rate can be significantly improved with increased bandwidth. Moreover, as the maximum transmit

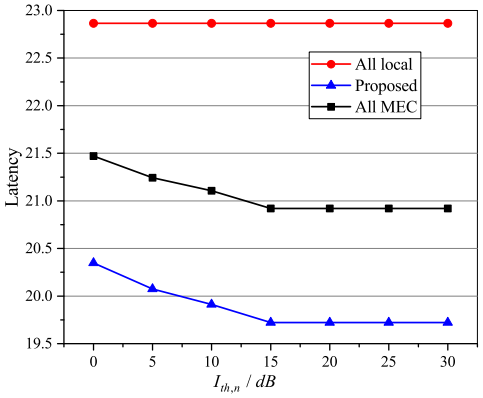

FIGURE 6. The impact of the interference threshold on network latency.

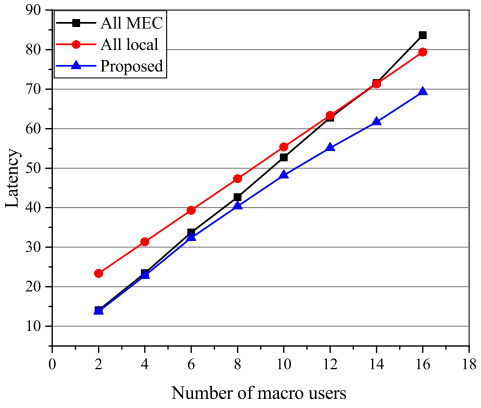

FIGURE 7. The effect of the number of macro-UEs on network latency.

power of the UEs increases, the weighted-sum execution latency can be further reduced. However, when the maximum transmit power reaches a certain threshold, the impact on network latency will decrease. This is because a high level of transmit power can improve the uploading rate. However, excessive transmit power can cause severe inter-user interference and limit the uploading rate for UEs.

Fig. 6 shows the impact of the interference threshold on the network latency obtained by the All MEC algorithm, the All local algorithm, and the proposed algorithm. As the interference threshold improves, the network latency of the All local algorithm does not change, but the network latency obtained by the All MEC algorithm and the proposed algorithm initially decrease and then remain unchanged. This is because the increase in the interference threshold improves the tolerance of co-channel interference generated by the UEs associated with the SBSs. Then, these UEs can increase the transmit power to improve the uploading rate. However, as the interference threshold is further improved, excessive co-channel interference limits the UEs' uploading rate.

Fig. 7 shows the effect of the number of macro-UEs on the network latency obtained by the All MEC algorithm, the All local algorithm, and the proposed algorithm. It can be concluded that, as the number of macro-UEs increases, the growth trend for the network latency obtained by the three algorithms will increase. However, the All MEC algorithm shows a faster growth trend than the All local algorithm. This is mainly because the average number of sub-channels obtained by the UEs decreases when the sub-channel resources are limited.

Fig. 8 shows the effect of the weight coefficients on the network latency obtained by the All MEC algorithm, the All local algorithm, and the proposed algorithm. To analyze the effect of the weight coefficients on network latency, we consider the case of two macro-UEs with $\omega_{0,1} + \omega_{0,2} = 1$ and no small cell UEs in the network. As seen in Fig. 8, the weight coefficients have no effect on the network latency obtained by the All local algorithm. For the All MEC algorithm and the proposed algorithm, fairness and priority between the UEs can be achieved by adjusting $\omega_{0,1}$ and $\omega_{0,2}$. In comparison to

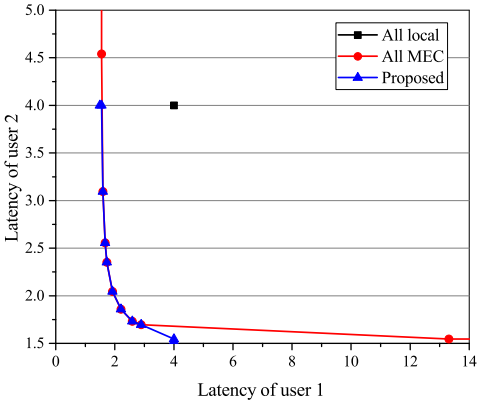

FIGURE 8. The effect of the weight coefficients on network latency.

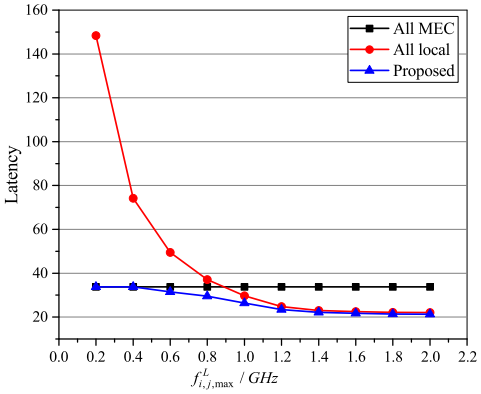

FIGURE 9. The effect of maximum local CPU frequency on the network latency.

the All MEC algorithm, the proposed algorithm can adjust the priority between UEs and also control the UEs' delay within a certain range, which depends on the local execution latency.

Fig. 9 and Fig. 10 show the results of the analysis of the effect of maximum local CPU frequency on the network latency and energy consumption. From the information presented in Fig. 8, we can conclude that the maximum local CPU frequency has no effect on the network latency obtained

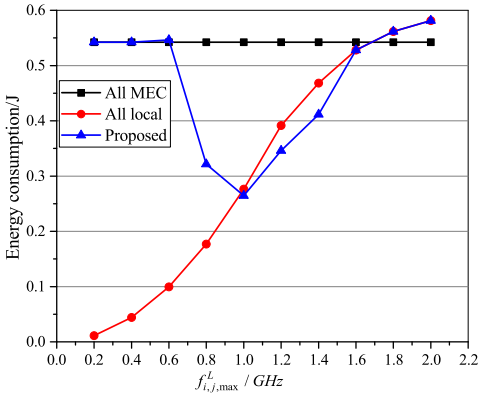

FIGURE 10. The effect of maximum local CPU frequency on the network energy consumption.

by the All MEC algorithm, and it only affects the network latency obtained by the All local algorithm and the proposed algorithm. As the maximum local CPU frequency increases, the network delay obtained by the All local algorithm gradually decreases, and eventually stabilizes. This is because the increase in the maximum local CPU frequency can drive more UEs to choose the local computing model. As seen in Fig. 9, as the maximum local CPU frequency increases, the overall energy consumption of the network tends to decrease at first, and then increase. There exists an optimal value. The reason for the decrease is that the increase in the maximum local CPU frequency can prompt more UEs to choose the local computing model. The increase in the local CPU frequency results in more energy consumption for each UE. As shown in Fig. 8 and Fig. 9, the appropriate maximum local CPU frequency needed for network latency and energy consumption is still an issue.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a joint computation offloading and resource allocation algorithm for IoT UEs in MEC-enabled HetNets. The simulation results demonstrate that the proposed algorithm has better network latency performance and computational complexity than the other evaluated algorithms. Moreover, the effects of the network parameters on network latency are analyzed using different algorithms. In future work, the network architecture of AI-enabled MEC systems deployed in HetNets will be further explored, and the optimal computation offloading decision for AI-enabled MEC systems will be also studied.

REFERENCES

[1] M. R. Rahimi, N. Venkatasubramanian, S. Mehrotra, and A. V. Vasilakos, "On optimal and fair service allocation in mobile cloud computing," *IEEE Trans. Cloud Comput.*, vol. 6, no. 3, pp. 815–828, Jul. 2018.

[2] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[3] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, 4th Quart., 2018.

[4] G. J. Sutton, J. Zeng, R. P. Liu, W. Ni, D. N. Nguyen, B. A. Jayawickrama, X. Huang, M. Abolhasan, Z. Zhang, E. Dutkiewicz, and T. Lv, "Enabling technologies for ultra-reliable and low latency communications: From PHY and MAC layer perspectives," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2488–2524, 3rd Quart., 2019.

[5] H. Guo, J. Liu, and J. Zhang, "Efficient computation offloading for multi-access edge computing in 5G HetNets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[6] 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.

[7] J. Xu, K. Ota, and M. Dong, "Saving energy on the edge: In-memory caching for multi-tier heterogeneous networks," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 102–107, May 2018.

[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] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.

[10] M. Qin, L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Power-constrained edge computing with maximum processing capacity for IoT networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4330–4343, Jun. 2019.

[11] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, Jun. 2019.

[12] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[13] Y. Pan, M. Chen, Z. Yang, N. Huang, and M. Shikh-Bahaei, "Energy-efficient NOMA-based mobile edge computing offloading," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 310–313, Feb. 2019.

[14] Z. Ding, D. W. K. Ng, R. Schober, and H. V. Poor, "Delay minimization for NOMA-MEC offloading," *IEEE Signal Process. Lett.*, vol. 25, no. 12, pp. 1875–1879, Dec. 2018.

[15] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial offloading scheduling and power allocation for mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6774–6785, Aug. 2019.

[16] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2651–2664, Dec. 2018.

[17] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[18] L. Lei, H. Xu, X. Xiong, K. Zheng, and W. Xiang, "Joint computation offloading and multiuser scheduling using approximate dynamic programming in NB-IoT edge computing system," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5345–5362, Jun. 2019.

[19] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[20] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[21] Z. Song, Y. Liu, and X. Sun, "Joint radio and computational resource allocation for NOMA-based mobile edge computing in heterogeneous networks," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2559–2562, Dec. 2018.

[22] Y. Tao, C. You, P. Zhang, and K. Huang, "Stochastic control of computation offloading to a helper with a dynamically loaded CPU," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1247–1262, Feb. 2019.

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

[24] 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.

[25] 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.
[26] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.
[27] W. Nam, D. Bai, J. Lee, and I. Kang, "Advanced interference management for 5G cellular networks," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 52–60, May 2014.
[28] V. W. S. Wong, *Key Technologies for 5G Wireless Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[29] CVX Research. (Aug. 2012). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.0*. [Online]. Available: http://cvxr.com/cvx
[30] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Oper. Res.*, vol. 26, no. 4, pp. 681–683, 1978.
[31] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[32] J. Zhang, E. Björnson, M. Matthaiou, D. W. K. Ng, H. Yang, and D. J. Love, "Multiple antenna technologies for beyond 5G," 2019, *arXiv:1910.00092*. [Online]. Available: http://arxiv.org/abs/1910.00092
[33] *Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for E-UTRA Physical Layer Aspects (Release 9)*, document TR 36.814 V9.0.0(2010.03), 3GPP, 2010.

LIANGRUI TANG received the Ph.D. degree in communication and information system from the Beijing University of Posts and Telecommunications. He is currently a Professor with the State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, focusing on the research of communication in power systems, wireless communications, and optical network communication. He has been a Reviewer of the IEEE TRANSACTIONS ON COMMUNICATION and the IEEE COMMUNICATIONS LETTERS.

HAILIN HU received the B.Sc. degree in telecommunications engineering from North China Electric Power University, in July 2016, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include convex optimization and resource allocation in 5G heterogeneous networks.