Offloading Schemes in Mobile Edge Computing With an Assisted Mechanism

HAOJIA WANG, ZHANGYOU PENG, AND YONGSHENG PEI

Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University, Shanghai 200444, China
Corresponding author: Zhangyou Peng (zypeng@shu.edu.cn)

ABSTRACT Mobile edge computing (MEC) is a promising paradigm for providing computing and storage capabilities in close proximity to mobile devices. To solve the scenario in which massive mobile devices have tasks to be processed at the same time, this paper proposes an assisted mechanism for the MEC system. When the primary MEC server is unable to meet the delay requirements of the mobile devices within its coverage area, a portion of the tasks can be offloaded to secondary MEC servers to obtain extra resources for processing. This MEC framework effectively reduces the computing and communication burden of the primary MEC server and improves the resource utilization of the secondary MEC servers. To maximize the system offloading utility in terms of latency, we formulated an optimization problem that jointly optimizes the task assignment, computing resource allocation and offloading decision of all mobile devices. Since the formulated problem is a mixed integer nonlinear problem, we use the decomposition method to convert the optimization problem into several subproblems. In addition, a heuristic algorithm based on the priorities of mobile devices and the MEC servers is proposed to obtain the suboptimal device offloading strategy. The numerical results show that the assisted mechanism can effectively reduce system latency and improve system reliability. In addition, the performance of our proposed algorithm is close to the optimal solution.

INDEX TERMS Assisted mechanism, computation offloading scheme, mobile edge computing, resource allocation.

I. INTRODUCTION

Recently, the rapid development of the Internet of Things has led to a significant increase in the number of mobile devices and application types [1]. Specifically, delay-sensitive and computing-intensive applications have the requirements of ultra-low latency, high reliability and low energy consumption for wireless access networks [2]–[5]. However, due to the limited computing capabilities and battery life of mobile devices, relying on local computing alone cannot meet the task requirements [6]. Thus, effectively implementing these mobile applications is a challenge. As a potential technology, mobile cloud computing (MCC) can provide sufficient computing and storage capability for mobile devices by offloading some tasks to a remote core cloud server (CCS) [7], [8]. However, because CCSs are usually far away from the mobile devices, this can cause long transmission delays and consume extra local energy during transmission, which may result in failure to complete the task within the specified time.

Mobile edge computing (MEC) is a promising paradigm that can effectively reduce the transmission latency caused by offloading tasks because it integrates cloud computing and communication capabilities providing considerable resources at the edge of the network (e.g., WIFI routers and gateways at home, micro data centers and cloudlets between the mobile devices and the core cloud [9], [10]). Therefore, MEC has received extensive attention from both academics and industry in recent years. The exact definition of MEC and related technical standards are provided by the European Telecommunications Standards Institute (ETSI) [11]. MEC is not only a key technology in the fifth generation (5G) mobile communication systems but will also play an important role in the next generation of mobile networks [12].

The emergence of MEC enables a large number of new applications and services (e.g., autonomous vehicles, face recognition, health monitoring and surveillance networks) to be effectively implemented by offloading their tasks to the MEC server [13], [14]. One of the most popular cases in 5G is virtual reality (VR) technology, which provides the closest personal experience to the screen through the transmission of various sights, sounds and emotions.
To achieve high-resolution video transmission at high frame rates, VR relies on MEC technology to migrate computation tasks from virtual reality devices to MEC servers with richer computing and storage resources. This is a useful method for improving the computing capability of local devices, and it saves their battery energy thus realizing an immersive experience [15].

Due to the limited computing resources of the MEC server, the computational capability of the MEC server is much smaller than that of the CCS. In addition, as the number of mobile devices increases, the competition of communication resources among mobile devices will also affect system performance [16]. In consideration of these disadvantages of the MEC server, it is not feasible to offload all tasks from mobile devices to the MEC server. Therefore, it is important to jointly optimize the computation offloading scheme for each mobile device to improve system performance in the MEC network.

**Motivation and Contributions:** To show the advantages of computing offloading in multiple mobile devices and multiple MEC servers scenario, we propose an assisted mechanism in the MEC system. Specifically, the secondary MEC servers can assist the primary MEC server to execute a portion of the offloading tasks. Furthermore, we optimize resource allocation, task assignment and offloading decision jointly to maximize the mobile devices’ offloading gains in our novel system. Thus, the problem is more challenging due to the nonnegligible intercell interference and coupling between multiple mobile devices. The main contributions of this paper are summarized as follows.

1) First, we formulate the optimization problem that maximizes the offloading utility in the system latency. Then, we jointly optimize the computing resource allocation, task assignment and device offloading decisions. Note that, offloading decisions include two aspects: whether the mobile device chooses to offload its computing task and which MEC server to select as the secondary MEC server.

2) The optimization problem is NP-hard, which makes it challenging to obtain the optimal solution. Therefore, we use the decomposition method to solve several subproblems that have low complexities. We fix the offloading decision of the mobile devices, and obtain the solutions of the computing resource allocation and task assignment problems by using convex optimization techniques and linear programming methods, respectively.

3) To reduce the complexity of the device offloading decision, we design a heuristic algorithm that can obtain the suboptimal solution according to the priorities of mobile devices and MEC servers.

4) Through a large number of numerical simulations under different strategies, we prove that the proposed algorithm can effectively reduce system latency and improve system reliability in the MEC system.

**Organization:** The rest of paper is organized as follows. In Section II, we review the related work. In section III, the system model is presented. The offloading utility in system latency and the optimization problem are formulated in Section IV. We proposed our approach to solve the problem in Section V and numerical results is presented in Section VI. Finally, conclusions are shown in Section VII.

**II. RELATED WORK**

In recent years, the MEC paradigm has attracted the attention of more and more researchers [17]. For instance, considering that a single mobile device has multiple tasks in MEC network, optimized task scheduling policies are obtained by using Markov decision process approach in [18], which can minimize the system latency and under the stringent constraint of energy consumption. The allocation of computing and communication resources are further considered in [19] to minimize the system latency and energy consumption, respectively. In addition, multiple mobile devices scenario is studied in [17], [20], [21]. Chen et al. [20] proposed a game theoretic method to obtain the efficient computation offloading schemes for each mobile device. Considering optimize the weighted sum of computation delay and energy, Zhang et al. [17] used the energy-aware offloading scheme to tradeoff these two merits in a multi-cell MEC framework. The similar problem is studied in [21], the complexity of the original optimization problem is reduced and the calculation accuracy is improved by using the submodule function. In recent years, a large number of works considered the collaboration of multiple MEC servers, which can effectively alleviate the impact of computing resource limitation on system performance. Specifically, Dinh et al. [22] proposed a novel system for a single mobile device, which can be assisted by multiple MEC servers to process its computing tasks, while other works extended this system to a multi-user scenario [1], [6], [10], [23]. In these works, optimizing the association between mobile devices and MEC servers is key to improving the quality of the user experience. Zhang et al. considered a two-tier computation offloading framework in the 5G heterogeneous networks, in which the Micro BS and the Small BS can execute the computation tasks collaboratively. Furthermore, the mobile device association decision in multi-cell scenario is studied in [10]. Except the vertical collaboration of multiple MEC servers, the horizontal collaboration among different mobile devices is studied in [24], [25], which uses the Device-to-Device (D2D) technology to enable massive mobile devices to share the computing and communication resources.

However, in the above works, the mobile device can only be associated with one MEC server, without the consideration that computing tasks can be assigned to different MEC servers. With the growth of mobile devices and task data size, the limited resources of the MEC server can seriously affect the performance of the system. Moreover, considering 5G ultra-dense cellular networks, the density of the base station (BS) has been increasing dramatically, which will result in a mobile device being in the range of multiple BSs with overlapping coverages [26]. Therefore, a mobile device
can offload its task to more than one nearby MEC server, which will effectively improve system performance.

Recently, Liu and Zhang [9] considered dividing a task into subtasks and offloading them sequentially to multiple MEC servers. Since the transmission and computation processes can be performed simultaneously on different MEC servers, the system latency can be effectively reduced. However, this work only considered a single mobile device, which ignores the competition of computing and communication resources among multiple mobile devices.

Difference with the above studies, we assume that, in a multiple mobile devices scenario, a device can associate with different MEC servers. To alleviate the computational burden of the MEC server and improve the utilization of nearby MEC servers, we propose an assisted mechanism in the MEC system that can effectively improve the performance of the system.

### III. SYSTEM MODEL

We consider a MEC system that consists of a primary base station (PBS) and multiple candidate secondary base stations (SBSs). Each BS is equipped with an MEC server, as shown in Fig. 1. Note that, in this paper, MEC server and BS can be interchanged. For instance, primary MEC server and PBS have the same meaning. MEC servers can provide computing resources to mobile devices and communicate with them through a wireless channel. A set of BSs is denoted as \( B = \{b_0, b_1, \ldots, b_I\} \), where \( b_0 \) is the PBS and \( \{b_1, \ldots, b_I\} \) are the candidate SBSs, which may have overlapping coverage areas with the PBS. If the available resources of the PBS cannot meet the task requirements, then SBSs can be selected by mobile devices to assist in completing a portion of its task. This framework consists of a set of mobile devices \( N = \{1, 2, \ldots, N\} \). Furthermore, we define the set of mobile devices \( M_i \) represents the set of mobile devices within the coverage of the BS \( b_i \), and these mobile devices are directly associated with the BS \( b_i \), that is known in advance. Each mobile device has a delay-sensitive and computation-intensive task to be executed. The mobile devices are randomly distributed within the coverage of BSs.

In order to reference easily, key parameters are introduced in Table 1.

#### TABLE 1. Key parameters.

| Notation | Definition |
|----------|------------|
| \( B \) | The set of BSs |
| \( N \) | The set of mobile devices |
| \( M_i \) | The set of mobile devices within the coverage of BS \( b_i \) |
| \( \tau_n \) | Computation task of mobile device \( n \) |
| \( c_n \) | Required CPU cycles for computing one bit of task \( \tau_n \) |
| \( v_n \) | Input data size of task \( \tau_n \) |
| \( f_n \) | Local computing capability of mobile device \( n \) |
| \( L \) | Local computing latency of task \( \tau_n \) |
| \( E_n \) | Energy consumed by mobile device \( n \) during local execution |
| \( E^C_n \) | Energy consumed by mobile device \( n \) when offloading its task |
| \( N_{i,\text{off}} \) | The set of mobile devices which offload their tasks to BS \( b_i \) |
| \( N_{o,\text{off}} \) | The set of mobile devices which offload their tasks in the system |
| \( R_i \) | The transmission rate from mobile device \( n \) to BS \( b_i \) |
| \( F \) | The mobile device offloading decision strategy |
| \( A \) | The task assignment scheme |

#### A. TASK AT THE LOCAL SIDE

In this network, each mobile device has one computation task to be completed that can be divided into subtasks, as in [19], [20], and [27]. Each computation task can be described as \( \tau_n = (v_n, c_n), n \in N \). Note that the parameters \( v_n \) and \( c_n \) can be obtained by carefully profiling of the task execution [16], [23]. For the task \( \tau_n \), \( v_n \) is the size of the input data, including input parameters, program codes, etc. [21], [23], and \( c_n \) represents the number of required CPU cycles for computing one bit of the task. Therefore, the workload of the task can be characterized by \( v_n c_n \), i.e., the total computation required to complete the task.

In our model, each mobile device can execute its tasks locally or offload them to MEC servers. We define \( f_n \) as the local computing capability of mobile device \( n \), which is measured by the number of CPU cycles per second [4]. Hence, the local execution time of task \( \tau_n \) can be given as

\[
 t_n^L = \frac{v_n c_n}{f_n}. \tag{1}
\]

The energy consumption of each CPU cycle can be denoted by \( \zeta(f_n^2) \) [7], where \( \zeta \) is the coefficient reflecting the chip architecture [5], [23]. Thus, the total energy consumption of the local execution can be calculated as

\[
 E_n^L = \zeta \left( f_n^2 \right)^2 v_n c_n. \tag{2}
\]

#### B. TASK AT THE MEC SIDE

In some cases, due to computing resources and channel capacity limitations, the PBS may not be able to handle all the offloaded tasks on its own. Thus, the mobile devices which in the set \( M_0 \) can offload a portion of their task data to their respective selected SBSs to lighten the computing burden of the PBS.
We assume that mobile devices establish communication with the BS using orthogonal frequency-division multiple access (OFDMA) [4], [17], [28]. Because each channel in the system is orthogonal to the others, the uplink intercell interference will not occur when the mobile devices transmit their tasks to the same BS. To improve spectrum utilization, we assume that each BS uses the same frequency band, where the interference between cells exists.

Therefore, when a mobile device $n$ chooses to access BS $b_i$ to offload a task, the signal-to-interference-plus-noise ratio (SINR) can be given by

$$\gamma_n = \frac{p_n h_{n, b_i}}{I_n + \delta^2},$$  

(3)

where $p_n$ denotes the transmitting power of a mobile device, and $h_{n, b_i}$ is the channel gain between mobile device $n$ and BS $b_i$, which represents the frequency-flat block-fading Rayleigh channels. The path loss can be modeled as $d_n^{-\gamma}$, where $d_{ni}$ and $\gamma$ denote the distance from the mobile devices $n$ to BS $b_i$ and the path loss exponent, respectively [19], [29]. $\delta^2$ is the background noise power. In addition, $I_n$ denotes the interference from the other cells, which can be calculated as

$$I_n = \sum_{k=1,k \neq n}^{N} \sum_{j=0,j \neq i}^{\ell} p_k h_{k,j}.$$  

(4)

Furthermore, we define $\gamma_n^{ij}$ as the task offloading indicator variable, where $\gamma_n^{ij} = 1$ means that the $n$th mobile device within the coverage of the BS $b_i$ and selects BS $b_j$ as its SBS. Note that, $j = i$ means this offloaded mobile device does not use an SBS to assist in the calculation. Note that $\gamma_n^{ij} = 0$ otherwise. We call BS $b_i$ and $b_j$ a pair of cooperative BSs.

The mobile device offloading decision strategy is defined as $\mathcal{Y} = \{\gamma_n^{ij} | \forall n \in \mathcal{N}, \forall b_i \in \mathcal{B}, \forall b_j \in \mathcal{B}\}$. We define $\mathcal{N}_{i,off} = \{n \in \mathcal{N} | \sum_{b_j \in \mathcal{B} \setminus b_i} \sum_{n \in \mathcal{N}} \gamma_n^{ij} = 1, \sum_{b_j \in \mathcal{B} \setminus b_i} \sum_{n \in \mathcal{N}} \gamma_n^{ji} = 1\}$ as the set of mobile devices that decides to offload their tasks to BS $b_i$. In addition, the total number of offloaded mobile devices in the system can be calculated by $\mathcal{N}_{off} = \{n \in \mathcal{N} | \sum_{b_j \in \mathcal{B}} \sum_{b_i \in \mathcal{B}} \sum_{n \in \mathcal{N}} \gamma_n^{ij} = 1\}$.

Hence, the achievable uplink transmission rate between mobile device $n$ and BS $b_i$ can be given by

$$R_n^i = \frac{W_i}{|N_{i,off}|} \log_2(1+\gamma_n^{ij}), \quad n \in \mathcal{N}.$$  

(5)

where $W_i$ is the bandwidth of BS $b_i$.

Mobile devices generally can be benefited from task offloading for the following two reasons. Firstly, the plenty of computing and communication resources in the MEC server can help mobile devices to overcome resource limitation. Secondly, by executing task remotely, the mobile devices could avoid spending a long time in high power state, thus save the energy and extend the battery life [30]. However, if all mobile devices transmit their tasks to the MEC server, a lower transmission rate will result [31]. Therefore, it is necessary to identify the optimal computation offloading scheme of all mobile devices in the system to balance the workload of the BSs and improve the uplink data rate and computing resource utilization.

1) TRANSMISSION DELAY

In the case where the mobile device chooses to offload its task to MEC servers, we denote $\alpha_n$ as the task assignment variable that represents the proportion of tasks offloaded to its directly associated BS

$$\alpha_n \in [0, 1], \quad n \in \mathcal{N}.$$  

(6)

Furthermore, we define the task assignment policy of all mobile devices as $\mathcal{A} = \{\alpha_n | n \in \mathcal{N}\}$. The size of the task transmitted to the directly associated BS $b_i$ can be presented as $v_n \alpha_n$. Hence, the transmission time can be calculated as

$$t_{upn}^i = \frac{v_n \alpha_n}{R_n^i}, \quad n \in M_i.$$  

(7)

Similarly, the transmission time of offloading the remaining task $v_n (1 - \alpha_n)$ to its corresponding SBS $b_j$ can be expressed as

$$t_{upn}^j = \frac{v_n (1 - \alpha_n)}{R_n^j}, \quad n \in M_i.$$  

(8)

2) COMPUTATION DELAY

The computing capabilities provided by the PBS and SBSs are shared by the offloading mobile devices. We denote the computing resource allocation strategy as $\mathcal{F} = \{f_n^i | n \in \mathcal{N}, b_i \in \mathcal{B}\}$, where $f_n^i$ is the amount of computing resources that BS $b_i$ allocates to mobile device $n$. Note that, if a mobile device chooses to execute the task at the MEC server, $\sum_{b_i \in \mathcal{B}} f_n^i > 0$, otherwise, $f_n^i = 0$.

Hence, the execution time of task $\tau_n$ at the cooperative BSs $b_i$ and $b_j$ can be calculated as

$$t_{exe}^{ij} = \frac{v_n \alpha_n e_n}{f_n^i}, \quad$$  

(9)

and

$$t_{exe}^{ij} = \frac{v_n (1 - \alpha_n) c_n}{f_n^j}.$$  

(10)

When mobile device $n$ offloads its task to MEC servers, the total experience time consists of three parts: the time takes to transmit the input data on the uplink, the computation delay at MEC servers and the time it takes to transfer the output data from MEC servers to the mobile device. Similar to many articles [20], [23], [27], [32], we omit the third time above since the amount of output data is generally much smaller than the input data and the downlink transmission rate is much higher than that of the uplink.

Specifically, if the mobile device selects $b_i$ and $b_j$ as cooperative BSs to execute its tasks, the first step is to transfer a portion of input data to $b_i$, and the second step includes two parts, (i) executing the task at the $b_i$, (ii) transferring the remaining task to the $b_j$ and executing it. Note that these two parts can be carried out simultaneously.
Therefore, the total latency of mobile device $n$ when offloading its task can be expressed as
\[ t_n^C = t_{upn} + \max \left\{ t_{exec}^i, t_{upn}^i + t_{exec}^j \right\}. \tag{11} \]

The energy consumed through offloading can be calculated as
\[ E_n^C = p_n \left( \frac{v_n \alpha_n}{R_n^i} + \frac{v_n (1 - \alpha_n)}{R_n^j} \right). \tag{12} \]

To sum up, the total energy consumed by mobile device $n$ can be calculated as
\[ E_n = \left( 1 - \sum_{b \in \mathcal{B}} y_{n}^{ij} \right) s_n^b + y_{n}^{ij} \beta_n \left( \frac{v_n \alpha_n}{R_n^i} + \frac{v_n (1 - \alpha_n)}{R_n^j} \right). \tag{13} \]

### IV. PROBLEM FORMULATION

In general, the quality of experience (QoE) for mobile devices is reflected in the task execution time, energy consumption, etc. In our considered system, the mobile device has strict requirements for energy consumption, and the application is sensitive to delay. To improve the QoE, it is important to minimize the task completion time for all mobile devices while guaranteeing that their energy consumption requirements are met.

We consider the relative improvement in task completion time, which can be denoted as $\frac{t_t - t_n^C}{t_t}$. Hence, the offloading utility of the mobile device $n$ is calculated by
\[ T_n = \left( \frac{t_t - t_n^C}{t_t} \right) y_n^{ij}, \quad \forall n \in \mathcal{N}, \tag{14} \]

Clearly, compared to local computation, the more time saved by the offloading process, the higher the utility value.

Based on the offloading decision $\mathcal{Y}$, the task assignment $\mathcal{A}$ and the computing resource allocation $\mathcal{F}$, we define the system utility contains all the mobile devices’ offloading utilities, which can be denoted as
\[ U(\mathcal{Y}, \mathcal{A}, \mathcal{F}) = \sum_{n \in \mathcal{N}} \beta_n T_n. \tag{15} \]

We use $\beta_n$ to represent the importance coefficient of mobile device $n$.

The objective of jointly determining task assignment, resource allocation and device offloading decision is to maximize the system utility. Hence, the optimization problem is formulated as follows
\[
\begin{align*}
\max_{\mathcal{Y}, \mathcal{A}, \mathcal{F}} & \quad U(\mathcal{Y}, \mathcal{A}, \mathcal{F}) \tag{16a} \\
\text{s.t.} & \quad y_{n}^{ij} \in \{0, 1\}, \quad \forall n \in \mathcal{N}, b_i \in \mathcal{B}, b_j \in \mathcal{B}, \tag{16b} \\
& \quad \sum_{b \in \mathcal{B}} y_{n}^{ij} \leq 1, \quad \forall n \in \mathcal{N}, b_i \in \mathcal{B}, \tag{16c} \\
& \quad f_n^v > 0, \quad \forall n \in \mathcal{N}_{\text{on}}, b^v_i \in \mathcal{B}. \tag{16d}
\end{align*}
\]

The constraints in the optimization problem can be explained as follows: (16b) and (16c) reflect the fact that all mobile devices can choose to execute their tasks locally or transfer them to MEC servers, and each mobile device can select at most one SBS. Obviously, the offloading decision is a binary variable. (16d) and (16e) imply that the computing resources allocated by the MEC server to the mobile device must be positive, and the sum of the computing resources allocated must not exceed the computing capability of the MEC server. (16f) indicates the range of the task assignment variable, which is continuous. (16g) states the energy constraints.

### V. LATENCY-OPTIMAL COMPUTATION OFFLOADING SCHEME

The objective problem in (16) contains both discrete variables $y_{n}^{ij}$ and continuous variables $\{f_n^i, \alpha_n\}$. Hence, this problem is formulated as a mixed-integer optimization problem, which is NP-hard in general [19]. Thus, it is difficult to obtain an optimal solution. Hence, we use the decomposition method to transform the optimization problem into several subproblems. We then solve them separately, which requires low-complexity. This will be explained in detail below.

The system utility in (15) can be rewritten as
\[ U(\mathcal{Y}, \mathcal{A}, \mathcal{F}) = \sum_{n \in \mathcal{N}} \beta_n y_n^{ij} - \sum_{n \in \mathcal{N}} \beta_n t_n^C y_n^{ij}. \tag{17} \]

First, fixing the mobile device offloading decision $\mathcal{Y}$, we find that only the second item on the right-hand side of (17) contains unknown variables. Therefore, the first subproblem is formulated as follows including the variables of task assignment and computing resource allocation
\[ U_1(\mathcal{A}, \mathcal{F}) = \sum_{n \in \mathcal{N}_{\text{off}}} \frac{\beta_n t_n^C}{t_n^C}. \tag{18} \]

Second, according to the optimal schemes of computing resource allocation $\mathcal{F}$ and task assignment $\mathcal{A}$, the optimal mobile device offloading decision can be obtained using the following second subproblem
\[ U_2(\mathcal{Y}, \mathcal{A}^*, \mathcal{F}^*) = \sum_{n \in \mathcal{N}} \left[ \beta_n - U_1(\mathcal{A}^*, \mathcal{F}^*) \right] y_{n}^{ij}. \tag{19} \]

### A. JOINT OPTIMIZATION OF COMPUTATION RESOURCE ALLOCATION AND TASK ASSIGNMENT

In this subsection, we will solve the optimization problem of computing resource allocation and task assignment. According to (16) and (18), the optimization function of the first subproblem can be expressed as follows
\[
\min_{\mathcal{A}, \mathcal{F}} U_1(\mathcal{A}, \mathcal{F}) \tag{20a}
\]
According to (1),\textsuperscript{11} we can transform the non-smooth problem into a smooth one to obtain a suboptimal solution. The basic idea is to transform the non-smooth problem into a smooth one to obtain a suboptimal solution. Therefore, the Lagrangian function of problem (22) is

\begin{equation}
L(\mathcal{F}, \lambda) = \sum_{n \in \mathcal{N}_{v_{n}, \text{off}}} \frac{\beta_{n} f_{n}^{\chi}}{f_{n}^{\nu}} + \lambda \left( \sum_{n \in \mathcal{N}_{v_{n}, \text{off}}} f_{n}^{\nu} - F_{v} \right).
\end{equation}

Firstly, calculate the derivation of the Lagrangian function with respect to \( f_{n}^{\nu} \), we can get

\begin{equation}
\frac{\partial L(\mathcal{F}, \lambda)}{\partial f_{n}^{\nu}} = -\frac{\beta_{n} f_{n}^{\chi}}{(f_{n}^{\nu})^{2}} + \lambda,
\end{equation}

\( (f_{n}^{\nu})^{*} \) satisfies the \( \frac{\partial L(\mathcal{F}, \lambda)}{\partial f_{n}^{\nu}} = 0 \), hence, \( (f_{n}^{\nu})^{*} \) can be calculated as follows

\begin{equation}
(f_{n}^{\nu})^{*} = \sqrt{\frac{\beta_{n} f_{n}^{\chi}}{\lambda}}.
\end{equation}

According to the KKT conditions, \( \lambda^{*} (\sum_{n \in \mathcal{N}_{v_{n}, \text{off}}} (f_{n}^{\nu})^{*} - F_{v}) = 0 \). Since \( \lambda^{*} > 0 \) is satisfied, we can get,

\begin{equation}
\sum_{n \in \mathcal{N}_{v_{n}, \text{off}}} (f_{n}^{\nu})^{*} - F_{v} = 0.
\end{equation}

Substituting (30) into (29), we can obtain the optimal solution of Lagrangian multiplier as follows

\begin{equation}
\lambda^{*} = \frac{\sum_{n \in \mathcal{N}_{v_{n}, \text{off}}} (\beta_{n} f_{n}^{\chi})}{F_{v}^{2}}.
\end{equation}

Then, substituting (31) into (29), the optimal computing resource allocation can be calculated by the following closed form

\begin{equation}
(f_{n}^{\nu})^{*} = F_{v} \sqrt{\frac{\beta_{n} f_{n}^{\chi}}{\sum_{n \in \mathcal{N}_{v_{n}, \text{off}}} (\beta_{n} f_{n}^{\chi})}}.
\end{equation}

Observing the equation above, we find that the mobile devices with poor computation capabilities will be allocated to fewer computing resources by the MEC server. Actually, this result is in line with our intuition since these mobile devices with poor computation capabilities will be allocated to fewer computing resources by the MEC server. Actually, this result is in line with our intuition since these mobile devices with poor computation capabilities will be allocated to fewer computing resources by the MEC server.

After obtaining \( (f_{n}^{\nu})^{*} \) in the above, we find that the optimization variables \( \{\alpha_{n}, t_{n}\} \), among the mobile devices, are independent and thus can be optimized separately. Therefore, we formulate the following optimization problem to obtain the task assignment scheme for each mobile device

\begin{equation}
\min_{t_{n}} \quad t_{n}^2 \quad \text{s.t.} \quad \alpha_{n} \in [0, 1], \quad \forall n \in \mathcal{N}, \quad G_{1} \left( \alpha_{n}, (f_{n}^{\nu})^{*} \right) \leq t_{n}, \quad (33b)
\end{equation}

\begin{equation}
G_{2} \left( \alpha_{n}, (f_{n}^{\nu})^{*} \right) \leq t_{n}, \quad (33d)
\end{equation}

\begin{equation}
p_{n} \left( \alpha_{n} v_{n} - \frac{(1 - \alpha_{n}) v_{n}}{R_{n}^{\nu}} \right) \leq E_{\text{max}}.
\end{equation}

\begin{equation}
\min_{t_{n}, \alpha_{n}} \quad t_{n} \quad \text{s.t.} \quad \alpha_{n} \in [0, 1], \quad \forall n \in \mathcal{N}, \quad G_{1} \left( \alpha_{n}, (f_{n}^{\nu})^{*} \right) \leq t_{n}, \quad (33b)
\end{equation}

\begin{equation}
G_{2} \left( \alpha_{n}, (f_{n}^{\nu})^{*} \right) \leq t_{n}, \quad (33d)
\end{equation}

\begin{equation}
p_{n} \left( \alpha_{n} v_{n} - \frac{(1 - \alpha_{n}) v_{n}}{R_{n}^{\nu}} \right) \leq E_{\text{max}}.
\end{equation}
It can be easily observed that the objective function and all the constraints in (33) are linear. Therefore, the optimal problem of task assignment is a linear programming problem of $\alpha_n$ and $t_n$, which we can solve using the graphic method [34].

According to the constraint (33c), $t_n$ is a monotonic increasing function about $\alpha_n$, and $t_n$ equals zero at the starting point $\alpha_n = 0$. The first derivation of $G_2 (\alpha_n)$ can be presented as follows

$$G_2 (\alpha_n) = \frac{\beta_n}{t_n} \left( \frac{v_n}{R_n} - \frac{v_n c_n}{f_n^*} - \frac{v_n}{R_n} \right), \quad (34)$$

Hence, the monotonicity of $t_n$ in (33d) is uncertain.

1) CASE 1

In this situation, we assume the following inequality is satisfied,

$$\frac{1}{R_n} \geq \frac{c_n}{f_n} + \frac{1}{R_n}, \quad (35)$$

Hence, $G_2 (\alpha_n) \geq 0$ and $G_2 (0) = \frac{\beta_n}{t_n} \left( \frac{v_n c_n}{f_n^*} + \frac{v_n}{R_n} \right) > 0$. This indicates that the constraint (33d) is monotonically increasing and is positive at the starting point. As shown in Fig. 2a. The shaded area in Fig. 2 represents the feasible domain of the function.

Note that, $\alpha_0$ is the intersection of $G_1 (\alpha_n)$ and $G_2 (\alpha_n)$, which can be denoted as

$$\alpha_0 = \frac{c_n R_n (f_n^*)^* + (f_n^*)^* (f_n^*)^*}{c_n R_n (f_n^*)^* + (f_n^*)^* (f_n^*)^* + c_n R_n (f_n^*)^*}. \quad (36)$$

Obviously, $\alpha_0$ is within the feasible domain of $\alpha_n$.

Observing the energy constraint in (33e), we can get the range of $\alpha_n$ as follows

$$\alpha_n \leq \frac{E_{\max} R_{nj} R_{ni} - p_n v_n R_{nj}}{p_n v_n (R_{nj} - R_{ni})}. \quad (37)$$

We denote the right-hand side of (37) as $\alpha_e$. If $\alpha_e > 0$, then the optimal problem in (33) is feasible; otherwise there is no feasible solution. In this paper, we assume that the feasibility condition is always satisfied. Specially, in the case 1, the optimal task assignment $\alpha_n^*$ and the minimum value of $t_n^*$ can be calculated as follows

$$\alpha_n^* = 0, \quad (38)$$

$$t_n^* = \frac{\beta_n v_n}{t_n} \left( \frac{c_n}{f_n^*} + \frac{1}{R_n} \right). \quad (39)$$

It implies that, if the SBS $b_j$ has the ability to provide relatively sufficient communication and computing resources, offloaded mobile devices tend to execute all their computing tasks on this SBS.

2) CASE 2

In the second situation, $\frac{1}{R_n} < \frac{c_n}{f_n^*} + \frac{1}{R_n}$. We have, $G_2 (0) = \frac{\beta_n}{t_n} \left( \frac{v_n c_n}{f_n^*} + \frac{v_n}{R_n} \right) > 0$. Therefore, the constraint in (33d) is monotonically decreasing and is positive at the starting point. As shown in Fig. 2b.

The optimal task assignment $\alpha_n^*$ in case 2 can be calculated as follows

$$\alpha_n^* = \min (\alpha_0, \alpha_e). \quad (40)$$

Finally, substituting the $\alpha_n^*$ into the left side of constraint (33d), we can calculate the optimal value of $t_n$ by,

$$t_n^* = \frac{\beta_n}{t_n} \left( \frac{\alpha_n^* v_n}{R_n} + \frac{1 - \alpha_n^*}{R_n} \left( \frac{c_n}{f_n^*} + \frac{1}{R_n} \right) \right). \quad (41)$$

B. OPTIMAL TASK OFFLOADING DECISION

In the previous section, computing resource allocation and task assignment schemes can be obtained according to (32), (38) and (40) by fixing the task offloading decision strategy. Next, we will solve the second suboptimal problem $U_2 (\mathcal{Y}, \mathcal{A}^*, \mathcal{F}^*)$ to obtain its solution. On the basis of (16) and (19), the optimization function can be expressed as follows

$$\max_{\mathcal{Y}} U_2 (\mathcal{Y}, \mathcal{A}^*, \mathcal{F}^*) \quad (42a)$$

s.t. $\gamma_n^j \in \{0, 1\}, \forall n \in \mathcal{N}, b_i \in \mathcal{B}, b_j \in \mathcal{B}, \quad (42b)$

$$\sum_{b_i \in \mathcal{B}} \gamma_n^j \leq 1, \forall n \in \mathcal{N}, b_i \in \mathcal{B}, \quad (42c)$$

$$E_n \leq E_{\max}, \forall n \in \mathcal{N}, b_i \in \mathcal{B}, b_j \in \mathcal{B}. \quad (42d)$$

It is quite challenging to solve the optimal problem in (42) due to the coupling of different mobile devices. Specifically, obtaining the optimal solution in polynomial time is extremely difficult. The simplest method is to use an exhaustive search to obtain the optimal solution. However, this method should consider all possible mobile device offloading decisions and SBS selection decisions. The complexity of this algorithm is $O \left( \left( I - 1 \right)^{M_0} \cdot 2^N \right)$, which is impractical.

To reduce the complexity of the algorithm, we design a heuristic algorithm to obtain the suboptimal solution of the offloading decision scheme. Thus, to reduce the latency of
the offloading system and improve the MEC servers’ utilization efficiency, the mobile devices and all candidate SBSs in our system should be assigned different priorities. First, we present two key definitions as follows,

Definition 1: The priority of SBS $b_i$ for mobile device $n$ is defined as

$$y_{ni} = \kappa_1 \left( \frac{1}{|\hat{M}|} M \right) + \kappa_2 \left( F_i \frac{E_f}{F} \right) + \kappa_3 \left( \frac{1}{Y_n^{\gamma_n}} \right), \quad (43)$$

where $\hat{M} = \sum_{i=1}^I \frac{1}{\gamma_i}$, $\hat{F}_s = \sum_{i=1}^{I} F_i$, and $\hat{Y}_n = \sum_{i=1}^{I} Y_n^\gamma_i$, $\kappa_\omega (\omega = \{1, 2, 3\})$ is the weight coefficient, which satisfies the following constraints: $0 \leq \kappa_\omega \leq 1$ and $\kappa_1 + \kappa_2 + \kappa_3 = 1$.

The first definition is used to select the optimal SBS for mobile devices. As shown in (43), the candidate SBS with low computational load, high computing capability and good channel quality tends to be selected first.

Definition 2: The priority of device $n$ in the process of offloading is defined as

$$y_n' = \psi_1 \left( \gamma_n \frac{1}{\hat{\gamma}} \right) + \psi_2 \left( \frac{1}{F_n} \hat{F}_n \right) + \psi_3 \left( E_n^{\gamma_c} - E_n^{\gamma_c} \frac{E}{E} \right) + \psi_4 (\hat{d}), \quad (44)$$

where $\gamma_n = \frac{\gamma_n + \gamma_M}{2}$, $\hat{\gamma} = \sum_{n \in \mathcal{N}} \gamma_n$, $\hat{F} = \sum_{n \in \mathcal{N}} \frac{1}{F_n}$, and $\hat{E} = \sum_{n \in \mathcal{N}} (E_n^{\gamma_c} - E_n^{\gamma_c})$. We denote the $\hat{d}$ as the indicator, such that, if the mobile device can be assisted by an SBS, $\hat{d} = 1$, otherwise, $\hat{d} = 0$. $\psi_i (i = \{1, 2, 3, 4\})$ is the weight coefficient, which satisfies the following constraints: $0 \leq \psi_i \leq 1$ and $\psi_1 + \psi_2 + \psi_3 + \psi_4 = 1$. Note that the priorities of the mobile devices and SBSs are proportional to $y_n'$ and $y_n$, respectively.

The offloading decision not only depends on the channel state between the mobile device and the MEC server but also on the location of the mobile device, the local computing capability and the energy consumption. Hence, we consider these four major factors in the second priority function. Specifically, the first item in (44) shows the effect of the channel state on mobile device priority. A mobile device with a higher SINR in the corresponding transmission channel should have higher priority since it can achieve a higher transmission rate. The second item indicates the effect of the local computing capability on the priority, and a mobile device that has a lower computing capability should be considered for offloading first. According to the third item in the formulation, mobile devices that save more computing energy by offloading should have higher priorities. The last item means that the mobile devices which can get the assistance of SBS should have higher priority since these mobile devices have the ability to offload a portion of their tasks to the SBS, thereby obtaining a higher offloading gain in the system.

The process of determining the mobile device offloading decision is presented in algorithm 1. The worst case complexity of this heuristic algorithm is $O(M_0 \cdot I + N^2)$. Obviously, the proposed heuristic algorithm can significantly reduce the complexity of the algorithm compared with the exhaustive method.

### Algorithm 1 Offloading Decision

**Require:**

$$\gamma_n, f_n, d_n, \mathcal{N}, \forall n \in \mathcal{N};$$

1: for all $n \in M_0$ do
2: for all $b_i \in B$ do
3: calculate priority $y_{ni}$ of SBS $b_i$ according to (43);
4: end for
5: select the BS with the highest priority as the SBS of the nth mobile device;
6: end for
7: for all $n \in \mathcal{N}$ do
8: calculate priority $y_n'$ of mobile device $n$ according to (44);
9: end for
10: Sort priority values for all mobile devices;
11: $U^*_2 (Y, A^*, F^*) = -\infty$;
12: $N^*_\text{off} = \phi$;
13: for $m = 0$ to $N$ do
14: generate a set $N_{\text{off}}$ that includes the $m$ mobile devices with the highest priorities;
15: calculate the objective value $U_2 (Y, A^*, F^*)$ of the mobile devices in $N_{\text{off}}$;
16: if $U^*_2 (Y, A^*, F^*) < U_2 (Y, A^*, F^*)$ then
17: $U^*_2 (Y, A^*, F^*) \leftarrow U_2 (Y, A^*, F^*)$;
18: $N^*_\text{off} \leftarrow N_{\text{off}}$
19: end if
20: end for

**Ensure:**

$N^*_\text{off}, U^*_2 (Y, A^*, F^*)$

First, the mobile device selects the BS with the highest priority as its SBS according to (43), and calculates the priority for the mobile devices using (44). The priority of different mobile devices is then sorted from high to low, and the mobile device with the highest priority tends to be offloaded first. The third step is to determine the number of offloaded mobile devices. The main idea is that, varying the number of the offloaded mobile devices $|N_{\text{off}}|$ from 0 to $N$, the optimal offloading strategy is to obtain the maximum value of $U_2 (Y, A^*, F^*)$. For instance, if $|N_{\text{off}}| = m (m \in [0, N])$, the mobile devices with the highest $m$ priority values will be selected to execute their tasks at MEC servers.

### VI. NUMERICAL RESULTS

Some simulation results is presented in this section to verify the system performance of our proposed latency-optimal computation offloading scheme (LOCOS). In this system, multiple mobile devices are distributed randomly at the circle with the BS as the center and radius 120m. The distance between each BS is set to 60m. And the uplink channel gain are obtained by using Rayleigh fading channel model, which can be expressed as follows,

$$h_n^i = \theta_0 d_n^{-\gamma} h_n^i, \quad (45)$$
We assume $\bar{h}_i^j$ obeys an independent and identically distributed (i.i.d.) circularly symmetric complex Gaussian (CSCG) random distribute, i.e. $\bar{h}_i^j \sim CN(0, I)$. We set the $\theta_0 = 6.25 \times 10^{-4}$, and the path-loss exponent $v$ is assumed to be 3 [35]. We get the numerical results through averaging 500 randomized realizations.

Unless otherwise state, we consider the MEC system includes a PBS and two SBSs. The number of mobile devices in the coverage of each SBS is set to 2. The transmission power of mobile device $p_n$ is set to 0.6W, and the background noise-power is $\delta^2 = 10^{-18}$. In addition, the system bandwidth of all BSs is 20MHz, the number of sub-bands allocates by each BS is equal to the total number of mobile devices associated with it.

From the perspective of computation task, we choose the data size and the number of required CPU cycles for computing one byte of each task are $v_n = 1.7$ MB and $c_n = 1900$ cycles/byte [36]. For computing resources, the CPU computation capability of MEC server is 10GHz, and the computation capabilities of mobile devices follow the uniform distribution with $f_n^j \sim U [0.6, 1]$ GHz. In the simulations, we set the energy coefficient $\varsigma$ as $5 \times 10^{-22}$ [23], and the importance coefficient of mobile device $\beta_n = 1$.

We present the following four benchmark polices to verify the performance of our algorithms.

- **All Offloading Computing (AOC):** All computing tasks of mobile devices are offloaded to the MEC server to execute. In this policy, we decide the optimal computing resource allocation, task assignment and SBS selection jointly.
- **All Local Computing (ALC):** Tasks for all mobile devices are executed locally. This scheme sets the variable $y_{ij} = 0, \forall n \in N, b_i \in B, b_j \in B$.
- **Offloading Scheme Without Cooperation Between PBS and SBSs (OSWC):** In MEC system, mobile devices decide their offloading decision and computing resource allocation independently without considering assisted mechanism.
- **Exhaustive:** As expressed in Section V part B, we use the exhaustive searching method to find the optimal offloading decision for all mobile devices. This method is used to compare the performance of our proposed strategy.

Improving the system offloading utility means reducing system latency as much as possible by the appropriate computation offloading schemes. Therefore, we use system latency as one of the performance indicators of the system.

Fig. 3 indicates the system latency under different number of mobile devices in the coverage of PBS by using above first three schemes and our proposed algorithm (LOCOS) respectively. We vary the number of mobile devices from 1 to 30. It can be found that, the system latency corresponding to these four policies increases as the number of mobile devices grows. And when the number is small, AOC algorithm and LOCOS algorithm have similar performance, since in this case, the computing capability and communication resources of BSs can meet the needs of mobile devices, all mobile devices tend to offload their tasks. However, with the number of mobile devices increases, our proposed LOCOS algorithm performs best. In addition, it can be observed that the performance of LOCOS algorithm is better than that of oswc algorithm, this because with the assisted mechanism, SBSs cooperatively execute some computing tasks of mobile devices, so the proposed system can make full use of the computing resources of SBSs and release the burden of PBS.

To verify the sub-optimality of the solution obtained by Algorithm 1, we compare it with the performance of the optimal solution, which is obtained by the exhaustive algorithm and then compare it with the OSWC algorithm. Considering the long running time of exhaustive method, we only consider a small number of mobile devices from 1 to 12 in the coverage area of PBS. According to the Fig. 4, it can be found that the performance of our proposed LOCOS algorithm is close to the optimal solution, and it is obviously better than the OSWC strategy. Fig. 5 shows the impact of input data size on system...
FIGURE 5. The system latency with different input data size.

FIGURE 6. The system latency with different computing resource allocation schemes.

FIGURE 7. Number of devices in four sets with different computing capability of SBS.

FIGURE 8. System reliability under different tolerable latency. $v_n = 0.6$ MB.

In order to characterize the performance of the proposed computing resource allocation strategy, we use the average resource allocation (ARA) method for all offloaded mobile devices as the benchmark for comparison. From Fig. 6, we observe that the proposed strategy of computing resource allocation is beneficial to improve system performance.

Fig. 7 shows the number of devices in the following four sets when the computing capability of SBSs is different, including (i) Offloading via the PBS only (PBS); (ii) Offloading via SBS only (SBS); (iii) Offloading via PBS and SBS coordinately (Coordinate); (iv) Computing on local device (Local). According to the Fig. 7, as the CPU computing capability of SBSs increases, mobile devices tend to offload their tasks to MEC servers rather than executing them locally. It can be explained that the execution time of MEC server will be reduced as the limitation of computing resources is relaxed, so mobile devices can improve system performance by offloading their tasks to remote MEC servers.

Next, we compare the performance of the system reliability by using LOCOS and OSWC algorithm respectively. Note that, we take the probability of measuring the maximum number of tasks can be completed in the hard-latency constraint as the index of system reliability. Fig. 8 shows that, with the tolerable latency relaxed, the probability of meeting the maximum latency constraint increases. And the proposed system has better performance than the system without SBSs cooperation, because the assisted mechanism can make use of the extra computing resources provided by SBSs, and alleviate the communication congestion caused by the increase of the number of mobile devices.

Finally, we highlight the influence of the number of SBSs on system total latency. As Fig. 9 indicates, with the increase latency with different offloading schemes, the number of mobile devices under the coverage area of PBS is fixed to 30. As it is evident, the system latency of each policy increases linearly with the size of the input data, and the solution obtained by LOCOS algorithm has substantial advantage in reducing system latency.
of the number of SBSs, the system latency decreases. It’s intuitive, with the increase of the number of SBSs, more computing tasks of mobile devices within the PBS can be executed with the assistance of SBSs, thus reducing the latency of the system.

VII. CONCLUSION

In this paper, we proposed an assisted mechanism for an MEC system, in which the PBS and SBSs can execute computation tasks collaboratively. To reduce the latency of the system, we formulated a NP-hard optimization problem to maximize the system offloading utility via optimizing the computing resource allocation, task assignment and offloading decision. We solve this problem by using a decomposition method to transform the original problem into several subproblems. In addition, we design a heuristic algorithm to obtain the suboptimal solution of the mobile device offloading decision, which is similar to the optimal solution obtained by the exhaustive method. Finally, we evaluated the performance of the system via simulations. The numerical results show that the assisted mechanism proposed in this paper has great performance in terms of system latency and reliability.

REFERENCES

[1] K. Cheng, Y. Teng, W. Sun, A. Liu, and X. Wang, “Energy-efficient joint offloading and wireless resource allocation strategy in multi-MEC server systems,” in *Proc. IEEE Int. Conf. Commun.* (ICC), Kansas City, MO, USA, May 2018, pp. 1–6.

[2] H. Tang, W. Zhang, W. Hardjawana, and B. Vucetic, “Improving latency and reliability in 5G Internet-of-Things networks,” in *Proc. IEEE Int. Conf. Smart Grid Commun.* (SmartGridComm), Sydney, NSW, Australia, Nov. 2016, pp. 509–513.

[3] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, “A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing,” *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1294–1313, 3rd Quart., 2013.

[4] 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.

[5] 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.

[6] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, “Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks,” *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

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

[8] C. Wang, C. Dong, J. Qin, X. Yang, and W. Wen, “Energy-efficient offload- ing policy for resource allocation in distributed mobile edge computing,” in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 366–372.

[9] 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.

[10] Y. Dui, D. Xu, S. Maharjan, and Y. Zhang, “Joint computation offloading and user association in multi-task mobile edge computing,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12313–12325, Dec. 2018.

[11] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, and A. Neal, “Mobile-edge computing—Introductory technical white paper,” Mobile-Edge Comput., ETSI, Sophia Antipolis, France, White Paper, Sep. 2014, no. 1, pp. 1089–7801. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%202018-09-14.pdf.

[12] J. Pan and J. McElhannon, “Future edge cloud and edge computing for Internet of Things applications,” *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, Feb. 2018.

[13] Y. X. Tran, M.-P. Hosseini, and D. Pomplii, “Mobile edge computing: Recent efforts and five key research directions,” *IEEE COMMOSC MMTIC Commun. Frontiers*, vol. 12, no. 4, pp. 29–33, 2017.

[14] J. Huang, C.-C. Xing, and C. Wang, “Simultaneous wireless information and power transfer: Technologies, applications, and research challenges,” *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 26–32, Nov. 2017.

[15] E. Bastug, M. Bennis, M. Medard, and M. Debbah, “Toward interconnected virtual reality: Opportunities, challenges, and enablers,” *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 110–117, Jun. 2017.

[16] L. Yang, J. Cao, H. Cheng, and Y. Ji, “Multi-user computation partitioning for latency sensitive multi-access cloud applications,” *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2253–2266, Aug. 2015.

[17] 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. 2636–2645, Aug. 2018.

[18] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, “Delay-optimal computation task scheduling for mobile-edge computing systems,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1451–1455.

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

[20] 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.

[21] Y. 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.

[22] T. Q. Thinh, 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.

[23] T. X. Tran and D. Pomplii, “Joint task offloading and resource allocation for multi-server mobile-edge computing networks,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[24] Y. He, J. Ren, G. Yu, and Y. Cai, “D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1750–1763, Mar. 2019.

[25] Y. He, J. Ren, G. Yu, and Y. Cai, “Joint computation offloading and resource allocation in D2D enabled MEC networks,” in *Proc. IEEE Int. Conf. Commun. (ICCC)*, Shanghai, China, May 2019, pp. 1–6.

[26] X. Ge, S. Tu, G. Mao, and C. X. Wang, “5G ultra-dense cellular networks,” *IEEE Trans. Wireless Commun.*, vol. 26, no. 4, pp. 974–983, Apr. 2017.

[27] D. Huang, P. Wang, and D. Niyato, “A dynamic offloading algorithm for mobile computing,” *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[28] C. Yui Wong, R. S. Cheng, K. B. Lataief, and R. D. Murch, “Multiuser OFDM with adaptive subcarrier, bit, and power allocation,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 10, pp. 1747–1758, Oct. 1999.
HAOJIA WANG received the B.E. degree in communication and information engineering from Shanghai University, Shanghai, China, in 2018, where she is currently pursuing the M.E. degree with the School of Communication and Information Engineering.

Her current research interests include the Internet of Things and mobile edge computing.

ZHANGYOU PENG received the M.S. degree in communication engineering from Tsinghua University, in 1992, and the Ph.D. degree in communication engineering from Shanghai University, in 2009.

Since 2002, he has been a Professor with the School of Communication and Information Engineering, Shanghai University. His research interests include satellite signal processing, radar signal processing, and wireless communication.

YONGSHENG PEI received the B.S. degree in communication engineering from Shanghai University, in 2018. He is currently pursuing the M.S. degree with Shanghai University.

His research interests include multiaccess edge computing, wireless communication theory, and convex optimization.

---

[29] 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.

[30] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, “A game theoretic resource allocation for overall energy minimization in mobile cloud computing system,” in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Jul. 2012, pp. 279–284.

[31] K. Kumar and Y.-H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[32] C. Xian, Y.-H. Lu, and Z. Li, “Adaptive computation offloading for energy conservation on battery-powered systems,” in *Proc. Int. Conf. Parallel Distrib. Syst. (ICDCS)*, Dec. 2007, pp. 1–8.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[34] K. G. Murty, *Linear Programming*. New York, NY, USA: Wiley, 1983.

[35] F. Wang, J. Xu, X. Wang, and S. Cui, “Joint offloading and computing optimization in wireless powered mobile-edge computing systems,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[36] M. Mukherjee, S. Kumar, M. Shojaifar, Q. Zhang, and C. X. Mavromoustakis, “Joint task offloading and resource allocation for delay-sensitive fog networks,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–7.