Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8288836, 11 pages
https://doi.org/10.1155/2021/8288836

Research Article

Optimization Strategy of Task Offloading with Wireless and Computing Resource Management in Mobile Edge Computing

Xintao Wu[1], **Jie Gan**[2], **Shiyong Chen**[1], **Xu Zhao**[2] and **Yucheng Wu**[1]

[1]*School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, China*
[2]*Beijing Smart-Chip Microelectronics Technology Co., Ltd., China*

Correspondence should be addressed to Shiyong Chen; chensy@cqu.edu.cn

Received 12 August 2021; Accepted 9 October 2021; Published 11 November 2021

Academic Editor: Dr. Saba Bashir

Mobile edge computing (MEC) provides user equipment (UE) with computing capability through wireless networks to improve the quality of experience (QoE). The scenario with multiple base stations and multiple mobile users is modeled and analyzed. The optimization strategy of task offloading with wireless and computing resource management (TOWCRM) in mobile edge computing is considered. A resource allocation algorithm based on an improved graph coloring method is used to allocate wireless resource blocks (RBs). The optimal solution of computing resource is obtained by using KKT conditions. To improve the system utility, a semi-distributed TOWCRM strategy is proposed to obtain the task offloading decision. Theoretical simulations under different system parameters are executed, and the proposed semi-distributed TOWCRM strategy can be completed with finite iterations. Simulation results have verified the effectiveness of the proposed algorithm.

1. Introduction

With the continuous development of the Internet of things and ubiquitous computing, mobile devices are increasingly running resource-intensive applications, such as interactive games and augmented reality [1, 2]. However, the limited resources of mobile devices cannot fully meet the requirements of these applications for powerful computing power and high speed. In recent years, many solutions have been proposed to solve the problem. In particular, mobile edge computing (MEC) provides a new way for UEs to complete computing tasks. MEC allows user equipment (UE) to offload computing tasks to network edge nodes through the wireless cellular network and performs the offloading tasks. This not only satisfies the expansion demand of users' computing capabilities but also compensates for the long delay of cloud computing [3]. It is a good method by using small base stations (SBSs) to meet the data rate demand of applications [4, 5]. As one of the key components of 5G, SBSs can enhance the coverage of local hot spots and increase system capacity. Dense network deployment can improve spectrum utilization and reduce end-to-end delay [6, 7].

However, task offloading not only generates additional overhead but also may cause intercell interference as it shares the same wireless frequencies among small cells, which will significantly influence the performance of the network [8]. Therefore, a reasonable offloading decision and interference management become the key to achieve efficient computation offloading [9]. A lot of works have been devoted to the research of computation offloading. Most of them have only focused on the process of offloading computing tasks from UE to MEC [10–18]. Only the optimal offloading decision is considered in [10, 11]. Researchers only focused on optimizing the communication resources [12, 13] or the computing resources [14, 15]. In some works, the combination of optimizing offloading decisions and resource allocation is used to minimize the latency or enhance the system performance [16–18]. Recently, research works by combining task offloading and interference management are proposed to improve the system utility [9, 19–21]. However, the scene of one user per base station is studied in [19, 20]. The work about wireless resource allocation does not take the minimum transmission rate requirement of each user into account [9].

The mobile devices can gather air quality data to analyze the environmental pollution or collect the image data to realize personal identity authentication from monitoring equipment. The MEC server determines whether the task is processed locally or offloaded to the server according to the computing capacity of the mobile device, the size of data, the delay, and the energy consumption requirements. The main contributions in this article are as follows:

(i) The communication model and the computing model in a multibase station and multiuser MEC scenario are described. The delay and energy consumption in local or remote computing are analyzed

(ii) The user utility is modeled as the weighted sum of the delay ratio and energy consumption ratio. And the system utility is defined as the sum of all user utilities. The optimization of the system utility is formulated by combining task offloading, wireless resource allocation, and computing resource allocation

(iii) The optimal goal is decomposed into three subproblems including wireless resource block allocation (RBA), computing resource allocation (CRA), and task offloading decision. The RBA is solved by using a resource allocation algorithm based on an improved graph coloring method. The optimal solution of CRA is obtained by using KKT conditions. In task offloading, a semi-distributed task offloading with wireless and computing resource management (TOWCRM) strategy is proposed to optimize the system utility under the constraints of computing resources

The rest of this article is organized as follows: the related works are discussed in Section 2. The system model with multiple cells and multiple users in the MEC scenario is described in Section 3. The optimization of the system utility is formulated in Section 4. In Section 5, wireless resource optimization and computing resource allocation are discussed. A semi-distributed TOWCRM algorithm is proposed to optimize offloading tasks. The simulation results are given and discussed in Section 6. The conclusion of this work is described in Section 7.

2. Related Works

Edge computing could be affected by external environment (such as wireless channel, interferences among mobile users, communication link quality, and the status of the communication channel) during offloading [22]. Therefore, it is very important to establish a suitable environment of offloading policy for computation offloading. In [10, 11], these studies only paid attention to task offloading without optimizing communication and computing resources. It was assumed that the capacity of cloud computing is unlimited, and some studies only focused on the optimization of communication resources in [12, 13]. For instance, to maximize the network management profit, an optimal solution algorithm based on the idea of branch-and-price was put forward to address joint resource management for device-to-device (D2D) communication [12]. Based on combining resource allocation and task assignment, a low-complexity iteration algorithm was proposed to minimize the task execution latency of all users subject to task and resource constraints in [13]. In contrast, only computing resource was optimized during task offloading [14, 15]. A new market-based framework was proposed to efficiently allocate computing resources of heterogeneous capacity-limited edge nodes (EN) for multiple competing services at the network edge in [14]. In [15], a smart contract that exploited the state-of-the-art machine learning algorithm was used in a private blockchain network to allocate the edge computing resources. In [16–18], joint communication and computing resource optimization were considered during the task offloading. To minimize the average latency of users to complete tasks, a strongly nonconvex problem with coupled variables was described as jointly considering the offloading decision, computation, and broadband resource allocation [16]. In [17], the problem of joint service caching, computation offloading, transmission, and computing resource allocation in a scenario of multiple users with multiple tasks was formulated to minimize the overall computation and delay costs. Moreover, the scenario where each user had a computation cost constraint was studied. A semi-distributed heuristic offloading decision algorithm (HODA) was proposed to maximize the system utility, which jointly optimized the offloading decision, communication, and computing resources [18].

In addition, there have been also some works that consider the joint optimization of task offloading and interference management at the same time [19–21]. Task offloading was studied in a MEC scenario with a single user per cell in [19, 20]. For example, offloading decision was made by considering the effect of intercell interference on system performance, where physical resource block (PRB) and computing resource allocation were treated as a joint optimization problem. The MEC server made the offloading decision to maximize the overhead, and the PRB was allocated by using a graph coloring algorithm [19]. In [21], the problem of joint task offloading and resource allocation was studied to maximize the offloading utility, which was modeled by the weighted sum of task completion time and device energy consumption. The resource allocation (RA) problem using convex and quasiconvex optimization was addressed, and a novel heuristic algorithm was proposed to solve the task offloading. It could achieve a suboptimal solution in polynomial time. However, there was no consideration to minimize interferences among mobile users.

3. System Model

This section describes the system model used in our work. Firstly, the network model is introduced in detail. Then, the corresponding communication model and calculation model are derived based on the proposed network model. For simplicity, the key notations used in the article are summarized in Table 1.

Table 1: Summary of key notations.

Notation	Description
$\mathcal{S}$	Set of SBSs
$\mathcal{U}_s$	Set of UEs in the coverage area of s
$\mathcal{X}$	The task offloading decision
$\mathcal{Y}$	The RB association strategy
$\mathcal{F}$	Computing resource allocation policy
$\mathcal{N}$	Set of RBs
B	The bandwidth of every RB
$x_{u_s^m}$	The offloading variable
$y_{u_s^m}^n$	RB assigned variable
$I_{u_s^m}^n$	The interference intensity
$P_{u_s^m}$	The transmission power of u_s^m
$K_{u_s^m}$	The number of RBs assigned to u_s^m
$H_{u_t^m,S}$	The channel gain between u_t^m and s
$R_{u_s^m}^r$	Uplink data rate from u_s^m to s
$\mathrm{CT}_{u_s^m}$	Computational task of u_s^m
$D_{u_s^m}$	Input data of computation task $\mathrm{CT}_{u_s^m}$
$C_{u_s^m}$	Workloads of computation task $\mathrm{CT}_{u_s^m}$
$f_{u_s^m}^{\mathrm{loc}}$	Local computing capability of u_s^m
$T_{u_s^m}^{\mathrm{loc}}$	Local execution time of task $\mathrm{CT}_{u_s^m}$
$T_{u_s^m,\mathrm{off}}^r$	Transmission time of task $\mathrm{CT}_{u_s^m}$ to the MEC server
$T_{u_s^m,\mathrm{exe}}^r$	Execution time of task $\mathrm{CT}_{u_s^m}$ at the MEC server
$E_{u_s^m}^{\mathrm{loc}}$	Energy consumption of u_s^m when executing its task locally
$E_{u_s^m,\mathrm{off}}^r$	Energy consumption of u_s^m when offloading its task $\mathrm{CT}_{u_s^m}$
$f_{u_s^m}^r$	Computing resources that the MEC server allocates to u_s^m
f	Computing resources of the MEC server
$\beta_{u_s^m}^t$	Preference of u_s^m on task completion time
$\beta_{u_s^m}^e$	Preference of u_s^m on task energy consumption
$W_{u_s^m}$	User utility of u_s^m
$\mathcal{O}_s$	The set of offloading UEs under each SBS
$R_{u_s^m}^{\min}$	The minimum rate requirement of u_s^m
$R_{1\times\mathcal{O}_s}$	User benefit matrix
$R_{R1\times N}$	Channel benefit matrix

3.1. Network Model. As shown in Figure 1, a two-layer cellular heterogeneous network composed of a macro cell base station (MBS) and S small cell base stations is considered [19, 20]. The MEC server is deployed on the side of the MBS and can perform multiple computing tasks at the same time. S SBSs are connected to the MEC server through optical fiber links like the MBS. Let $\mathcal{S} = \{1, 2, \cdots, s, \cdots, S\}$ be the set of SBSs, and there are M UEs associated with each SBS in its coverage. We denote the set of UEs in the coverage area of s as $\mathcal{U}_s = \{u_s^1, u_s^2, \cdots, u_s^m, \cdots, u_s^M\}$, where u_s^m represents a UE belonging to s. In addition, for simplicity, the mobility of users or the handover among cells was not considered as it was assumed in [23–25]. Similar to many previous works in cloud computing and mobile networks [26–28], it is a semistatic scenario, which means that the position and transmission channel conditions remain unchanged during offloading a task.

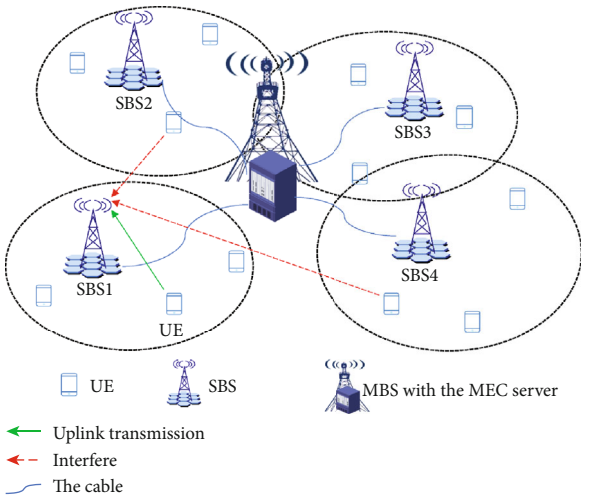


Figure 1: Cellular heterogeneous network model in mobile edge computing.

3.2. Communication Model. It is assumed that each UE has a time-sensitive task that requires a lot of computing resources to complete. Each UE can perform by offloading the computing task to the MEC server through its associated SBS or execute the computing task locally. Therefore, we denote the offloading variable as $x_{u_s^m} \in \{0, 1\}$. $x_{u_s^m} = 0$ means that u_s^m performs its task locally. $x_{u_s^m} = 1$ means that the user of u_s^m chooses to offload the task to the MEC server via a wireless link. The task offloading decision can be expressed as $\mathcal{X} = [x_{u_s^m}]$, which is a matrix of $S \times M$.

Uplink spectrum multiplexing is used in this model. The spectrum resources of the entire system are divided into N orthogonal RBs, and the RB set is defined as $\mathcal{N} = \{1, 2, \cdots, n, \cdots, N\}$ The RB associated table is defined as $Y_s = \{y_{u_s^m}^n\}$, which is a $M \times N$ matrix, where M is the total number of UEs in the s-th cell and N is the total number of RBs. $y_{u_s^m}^n = 1$ means that the n-th RB is assigned to u_s^m; otherwise, $y_{u_s^m}^n = 0$. And the RB allocation strategy is defined as $\mathcal{Y} = \{Y_s\}$, $s \in S$.

During uplink transmission, each UE and each SBS have a single antenna for sending and receiving messages. When u_s^m offloads its task to the MEC server for calculation, interference will occur if there are UEs in other SBSs sharing the same RB(s) with the current u_s^m . As RBs are assigned orthogonally to users in each cell, there is no interference

in intracell. The interference transmission power from u_t^m sharing the n-th RB to the s-th cell can be described as

$$I_{u_s^m}^n = \sum_{t=1,t\neq s}^{S} \sum_{m=1}^{M} x_{u_t^m} y_{u_s^m}^n \frac{P_{u_t^m}}{K_{u_t^m}} H_{u_t^m,s}, \tag{1}$$

where $P_{u_t^m}$ represents the transmission power of u_t^m, $K_{u_t^m}$ stands for the number of RBs assigned to u_t^m, and $H_{u_t^m,S}$ denotes the channel gain between u_t^m and s.

Given the decision matrix $\mathcal{X}$ and the RB associated strategy $\mathcal{Y}$, the uploading rate achieved by u_s^m connected to s can be obtained by Shannon's formula as [19]

$$R_{u_s^m}^r(\mathcal{X},\mathcal{Y}) = x_{u_s^m} \sum_{n=1}^{N} y_{u_s^m}^n B \log_2 \left(1 + \frac{P_{u_s^m} H_{u_s^m,s}}{K_{u_s^m}\left(I_{u_s^m}^n + \sigma^2\right)}\right), \tag{2}$$

where σ^2 is the variance of background noise, B is the bandwidth of each RB, $P_{u_s^m}$ represents the transmission power of u_s^m, $K_{u_s^m}$ stands for the number of RB allocated to u_s^m, and $H_{u_s^m,s}$ denotes the channel gain between u_s^m and s.

3.3. Calculation Model. The computing task of u_s^m is described as $\mathrm{CT}_{u_s^m} = <D_{u_s^m}, C_{u_s^m}>$, in which $D_{u_s^m}$ (in kB) represents the size of transmission data and $C_{u_s^m}$ (in megacycles) specifies the workload, i.e., the number of CPU cycles required to complete the computing task. The values of $D_{u_s^m}$ and $C_{u_s^m}$ can be obtained by carefully analyzing the offloading task [29, 30]. The delay and power consumption of local and remote computation will be discussed, respectively.

(1) Local computing: let $f_{u_s^m}^{\mathrm{loc}} > 0$ represent the local computing capacity of u_s^m in terms of the number of CPU cycles/s. The computation time $T_{u_s^m}^{\mathrm{loc}}$ for the local execution of the task $\mathrm{CT}_{u_s^m}$ can be expressed as

$$T_{u_s^m}^{\mathrm{loc}} = \frac{C_{u_s^m}}{f_{u_s^m}^{\mathrm{loc}}}, \tag{3}$$

and the energy consumption $E_{u_s^m}^{\mathrm{loc}}$ is denoted as

$$E_{u_s^m}^{\mathrm{loc}} = k\left(f_{u_s^m}^{\mathrm{loc}}\right)^2 C_{u_s^m}, \tag{4}$$

where $k(f_{u_s^m}^{\mathrm{loc}})^2$ is the energy consumption per calculation cycle and k depends on the energy coefficient on the chip architecture. According to the actual measurement, $k = 10^{-27}$ is usually adopted [21].

(2) Remote computing: u_s^m is connected to the corresponding s through a wireless network, and its task is offloaded to the MEC server for calculation. The computing resources provided by the MEC server are quantified by the computing capacity f (CPU cycles/s), which can be shared among the related UEs. The uplink transmission delay of u_s^m can be expressed as follows:

$$T_{u_s^m,\mathrm{off}}^r = \frac{D_{u_s^m}}{R_{u_s^m}^r(\mathcal{X},\mathcal{Y})}. \tag{5}$$

When a computing task $\mathrm{CT}_{u_s^m}$ is offloaded to the MEC server, the MEC server allocates specific computing resources to process the task, which is represented by $f_{u_s^m}^r$ (CPU cycles/s). And the computing resource allocation profile is defined as $\mathcal{F} = \{f_{u_s^m}^r\}$. During the execution of the task, it is assumed that the calculation speed assigned by the MEC server to each UE is fixed. The time of the MEC server executing the task is described as

$$T_{u_s^m,\mathrm{exe}}^r = \frac{C_{u_s^m}}{f_{u_s^m}^r}. \tag{6}$$

In addition, a feasible computing allocation strategy must satisfy the constraints of computing resources, which can be expressed as

$$\sum_{s\in\mathcal{S}} \sum_{u_s^m\in\mathcal{U}_s} x_{u_s^m} f_{u_s^m}^r \leq f. \tag{7}$$

The total delay of u_s^m for finishing the task is given by the following equation:

$$T_{u_s^m}^r = T_{u_s^m,\mathrm{exe}}^r + T_{u_s^m,\mathrm{off}}^r = \frac{C_{u_s^m}}{f_{u_s^m}^r} + \frac{D_{u_s^m}}{R_{u_s^m}^r(\mathcal{X},\mathcal{Y})}. \tag{8}$$

Through the above analysis, the energy consumption of u_s^m during the transmission data can be calculated as

$$E_{u_s^m,\mathrm{off}}^r = P_{u_s^m} \times T_{u_s^m,\mathrm{off}}^r = \frac{P_{u_s^m} D_{u_s^m}}{R_{u_s^m}^r(\mathcal{X},\mathcal{Y})}, \tag{9}$$

where $P_{u_s^m}$ represents the transmitting power of u_s^m.

We mainly consider the energy consumption and delay of UEs, and the computing energy consumption of the MEC server is omitted. As the amount of data returned to the mobile users is small, the power consumption and latency of UE receiving the returned data are omitted.

4. Problem Formulation

In this section, the problem of task offloading, wireless RBs, and computing resource allocation is formulated under the definition of user and system utility.

In a mobile cloud computing system, UEs' preference is mainly manifested in task completion time of $\beta_{u_s^m}^t$ and energy consumption of $\beta_{u_s^m}^e$. $\beta_{u_s^m}^t$, $\beta_{u_s^m}^e \in [0,1]$, and $\beta_{u_s^m}^t + \beta_{u_s^m}^e$

= 1. The quality of experience (QoE) can be described by comparing the delay and the power consumption of remote computing with that of local execution [18, 21]. The user utility of $W_{u_s^m}$ for u_s^m can be defined as

$$W_{u_s^m} = \left(\beta_{u_s^m}^t \frac{T_{u_s^m}^{\text{loc}} - T_{u_s^m}^r}{T_{u_s^m}^{\text{loc}}} + \beta_{u_s^m}^e \frac{E_{u_s^m}^{\text{loc}} - E_{u_s^m,\text{off}}^r}{E_{u_s^m}^{loc}} \right) x_{u_s^m}. \quad (10)$$

$\beta_{u_s^m}^t$ and $\beta_{u_s^m}^e$ can be determined according to the life of the remaining battery and the mission completion time requirements. From the above expression, it is clear that its user utility $W_{u_s^m}$ is equal to 0 when the task of u_s^m is executed locally ($x_{u_s^m} = 0$). When the task of u_s^m is executed on the MEC server ($x_{u_s^m} = 1$), its user utility $W_{u_s^m}$ is larger than 0.

Given the offloading policy of $\mathcal{X}$, the RB allocation strategy of $\mathcal{Y}$, and the calculating resource allocation policy of $\mathcal{F}$, the system utility can be defined as the sum of all user utilities and is expressed as follows:

$$W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) = \sum_{s\in\mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} W_{u_s^m}. \quad (11)$$

To maximize the system utility by jointly optimizing task offloading, wireless RBs, and computing resource allocation in mobile edge computing, the optimal goal can be formulated as

$$\begin{aligned} &\max_{\mathcal{X},\mathcal{Y},\mathcal{F}} W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) \\ &\text{s.t.} C1 : x_{u_s^m} \in \{0,1\} \forall u_s^m \in \mathcal{U}_s, s \in \mathcal{S} \\ &C2 : y_{u_s^m}^n \in \{0,1\} \forall u_s^m \in \mathcal{U}_s, n \in \mathcal{N}, s \in \mathcal{S}, \\ &C3 : \sum_{s\in\mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} x_{u_s^m} f_{u_s^m}^r \le f. \end{aligned} \quad (12)$$

The constraints in the above formula can be interpreted as follows: constraint $C1$ in (12) implies that the task can be executed locally or offloaded to the MEC server for execution. Constraint $C2$ in (12) indicates whether the n-th RB is assigned to u_s^m. Constraint $C3$ in (12) ensures that the sum of computing resources allocated to all offloading UEs does not exceed the computing capacity of the MEC server.

Due to the existence of integer variables, the above equation is a mixed integer nonlinear program (MINLP) problem [31]. The equation of (12) can be rewritten as follows:

$$\max_{\mathcal{X},\mathcal{Y},\mathcal{F}} W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) = \max_{\mathcal{X}} \left(\max_{\mathcal{Y},\mathcal{F}} W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) \right). \quad (13)$$

From (13), it can be seen that offloading decision, RB allocation, and computing resource allocation are decoupled from each other [32].

The original problem can be translated into offloading decision and resource allocations. In the next section, we will present solutions to both the resource allocations and task offloading decision.

5. Resource Optimization and Task Offloading Strategy

In this section, considering the time delay and energy consumption demand of UEs, a resource allocation algorithm based on improved graph coloring is used to allocate RBs. The solution of computing resources is obtained by using KKT conditions, and a semi-distributed TOWCRM algorithm is adopted to optimize the offloading decision.

The set of offloading UEs for the s-th SBS is defined as $\mathcal{O}_s$.

If a feasible task offloading decision is given, the objective function of (12) can be translated as follows:

$$\begin{aligned} &\max_{\mathcal{X},\mathcal{F}} W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) = \max_{\mathcal{Y},\mathcal{F}} \left(\sum_{s\in\mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} \left(\beta_{u_s^m}^t \beta_{u_s^m}^e \right) - V(\mathcal{X}, \mathcal{Y}, \mathcal{F}) \right), \\ &\text{s.t.} C1 : \mathcal{Y}_{u_s^m}^n \in \{0,1\} \forall u_s^m \in \mathcal{U}_s, n \in \mathcal{N}, s \in \mathcal{S}, \\ &C2 : \sum_{s\in\mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} x_{u_s^m} f_{u_s^m}^r \le f, \end{aligned} \quad (14)$$

where

$$V(\mathcal{X}, \mathcal{Y}, \mathcal{F}) = \sum_{s\in\mathcal{S}} \sum_{u_s^m \in \vartheta_s} \left(\frac{\beta_{u_s^m}^t T_{u_s^m}^r}{T_{u_s^m}^{\text{loc}}} + \frac{\beta_{u_s^m}^e E_{u_s^m,\text{off}}^r}{E_{u_s^m}^{\text{loc}}} \right). \quad (15)$$

From (14), it is easy to see that $\sum_{s\in\mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} (\beta_{u_s^m}^t + \beta_{u_s^m}^e)$ is an exact value for a specific offloading decision of $\mathcal{X}$. The $V(\mathcal{X}, \mathcal{Y}, \mathcal{F})$ can be regarded as the total offloading cost of all UEs who need to be offloaded. Therefore, the equation of (14) can be equivalent to minimize the total offloading overheads.

$$\begin{aligned} &\min_{\mathcal{Y},\mathcal{F}} V(\mathcal{X}, \mathcal{Y}, \mathcal{F}) = \min_{\mathcal{Y},\mathcal{F}} \left(\sum_{s\in\mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} \frac{\phi_{u_s^m} + \psi_{u_s^m}}{R_{u_s^m}^r(\mathcal{X}, \mathcal{Y})} + \sum_{s\in\mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} \frac{\eta_{u_s^m}}{f_{u_s^m}^r} \right) \\ &\text{s.t.} C1 : \mathcal{Y}_{u_s^m}^n \in \{0,1\} \forall u_s^m \in \mathcal{U}_s, n \in \mathcal{N}, s \in \mathcal{S}, \\ &C2 : \sum_{s\in\mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} f_{u_s^m}^r \le f, \end{aligned} \quad (16)$$

where $\varnothing_{u_s^m} = \beta_{u_s^m}^t D_{u_s^m} / T_{u_s^m}^{\text{loc}}$, $\psi_{u_s^m} = \beta_{u_s^m}^t D_{u_s^m} P_{u_s^m} / E_{u_s^m}^{\text{loc}}$, and $\eta_{u_s^m} = \beta_{u_s^m}^t f_{u_s^m}^{\text{loc}}$.

It can be seen from (16) that RB allocation and computing resource allocation are decoupled from each other in the target and constraint. We can decouple problem (16) into two independent problems, namely, resource block allocation (RBA) and computing resource allocation (CRA), and their respective solutions are presented in the following sections.

5.1. Resource Block Allocation (RBA). Taking the first term in (16) as the objective function, the RB assignment problem of $\Gamma(\mathcal{X}, \mathcal{Y})$ can be written as

$$\min_{\mathscr{Y}} \Gamma(\mathscr{X}, \mathscr{Y}) = \min_{\mathscr{Y}} \sum_{s\in\mathscr{S}} \sum_{u_s^m \in \mathscr{O}_s} \frac{\phi_{u_s^m} + \psi_{u_s^m}}{R^r_{u_s^m}(\mathscr{X}, \mathscr{Y})}$$
$$\text{s.t.} y^n_{u_s^m} \in \{0, 1\} \forall u_s^m \in \mathscr{U}_s, n \in \mathscr{N}, s \in \mathscr{S}. \tag{17}$$

Note that in the RB allocation phase, it is assumed that all UEs are transmitted with a fixed transmission power of $P_{u_s^m}$. The transmitted power of each UE is equally distributed over each RB assigned to it. From (17), the minimal value of $\Gamma(\mathscr{X}, \mathscr{Y})$ is obtained if the transmission rate of each offloading UE is maximized.

In order to better illustrate the transmission quality, the minimum transmission rate (when all UEs of the system are offloaded, computing resources are equally distributed to all UEs) is expressed as

$$R^{\min}_{u_s^m} = \frac{D_{u_s^m} \times \mathrm{f}}{T^{\mathrm{loc}}_{u_s^m} \times f - C_{u_s^m} \times S \times M}. \tag{18}$$

For the above RBA problem, it can be equivalent to the matching problem of the number of offloading UEs and the number of RBs. Therefore, a resource allocation algorithm based on an improved graph coloring method [19] is proposed to solve the above problem. The algorithm flow is simply described as follows:

(1) Initialization (step 1): in this step, the MEC server sets the RB allocation strategy $\mathscr{Y}$ to zeros and constructs S user benefit matrices as $R_{1\times\vartheta_s} = \{r_{u_s^m}\}$, where every user rate is $r_{u_s^m} = B\log_2(1 + (P_{u_s^m} H_{u_s^m,s}/\sigma^2))$, $u_s^m \in \vartheta_s, s \in \mathscr{S}$. At the same time, each UE also constructs its own channel benefit matrix $R\_R_{1\times N} = \{y^n_{u_s^m} B\log_2(1 + (P_{u_s^m} H_{u_s^m,s}/K_{u_s^m}(I^n_{u_s^m} + \sigma^2)))\}$, $u_s^m \in \vartheta_s, s \in \mathscr{S}$

(2) Orthogonal allocation (step 2): if the number of the offloading UEs is less than or equal to the number of RBs, RBs are allocated according to a uniform zero frequency reuse (UZFR) method [9]. Otherwise, the elements in S user benefit matrices are sorted by $r_{u_s^m}$ in descending order, and RBs are assigned to the first N UEs according to the UZFR method

(3) Allocate the RB with the greatest channel benefit (step3): on the basis of step 2, the MEC server selects a UE with the best user benefit from the remaining UEs. The selected UE starts to update its own channel benefit matrix according to the RB allocation strategy at this time and then selects the RB with the greatest channel benefit as the transmission RB. The MEC server deletes the selected UE from the remaining UEs. Step 3 will be repeated until all offloading UEs are assigned to RB

(4) Check whether all offloading UEs meet the minimum rate (step 4): according to (18), all offloading UEs will be checked whether they meet the minimum transmission rate. If satisfied, the algorithm terminates. If not, these UEs need to continue to allocate RBs. The set of these UEs is denoted as I'

(5) RB reallocation (step 5): the MEC server selects a UE with the best user benefit from UEs that needed to continue to allocate RBs. The selected UE starts to update its own channel benefit matrix according to the RB allocation strategy at this time and then selects the RB with the greatest channel benefit as the transmission RB. The MEC server deletes the selected UE from UEs that needed to continue to allocate RBs. Repeat step 5 until all UEs that do not meet the minimum rate are once again assigned to RB

(6) Iterative loop (step 6): step 4 to step 5 will be repeated until all UEs meet the minimum rate or $I' = \varnothing$. Then, the algorithm terminates and the RB allocation strategy $\mathscr{Y}^*$ is obtained under the offloading decision

The optimal objective function of (17) can be calculated as

$$\min \Gamma(\mathscr{X}, \mathscr{Y}) = \min \sum_{s\in\mathscr{S}} \sum_{u_s^m \in \mathscr{O}_s} \frac{\phi_{u_s^m} + \psi_{u_s^m}}{R^r_{u_s^m}(\mathscr{X}, \mathscr{Y}^*)}. \tag{19}$$

5.2. Computing Resource Allocation (CRA). From (16), computing resource allocation (CRA) is to optimize the second term of formula (16) and is expressed as follows:

$$\min_{\mathscr{F}} \phi(\mathscr{X}, \mathscr{F})$$
$$\text{s.t.} C1 : \sum_{s\in\mathscr{S}} \sum_{u_s^m \in \mathscr{O}_s} f^r_{u_s^m} \le f,$$
$$C2 : f^r_{u_s^m} > 0, \tag{20}$$

where

$$\Phi(\mathscr{X}, \mathscr{F}) = \sum_{s\in\mathscr{S}} \sum_{u_s^m \in \mathscr{O}_s} \frac{\eta_{u_s^m}}{f^r_{u_s^m}}. \tag{21}$$

From the above equation, it is a convex optimization problem. And constraint $C2$ in (20) is slack based on Karush-Kuhn-Tucker conditions, and it can be solved by using the KKT conditions.

The equivalent Lagrange function of this problem can be expressed as

$$L\left(f^r_{u_s^m}, \beta\right) = \sum_{s\in\mathscr{S}} \sum_{u_s^m \in \vartheta_s} \frac{\eta_{u_s^m}}{f^r_{u_s^m}} + \beta\left(\sum_{s\in\mathscr{S}} \sum_{u_s^m \in \vartheta_s} f^r_{u_s^m} - f\right). \tag{22}$$

Let $\beta > 0$ be the Lagrange operator; the derivatives of the Lagrange function of $L(f^r_{u_s^m}, \beta)$ can be described as

$$\frac{\partial L\left(f^r_{u^m_s}, \beta\right)}{\partial f^r_{u^m_s}} = -\frac{\eta_{u^m_s}}{\left(f^r_{u^m_s}\right)^2} + \beta. \tag{23}$$

By setting the above equation equal to 0, the solution of optimal computing resource allocation for problem (20) can be obtained.

$$\left(f^r_{u^m_s}\right)^* = \sqrt{\frac{\eta_{u^m_s}}{\beta}}, \tag{24}$$

where

$$\sum_{s\in\mathcal{S}}\sum_{u^m_s\in\vartheta_s}\left(f^r_{u^m_s}\right)^* = f. \tag{25}$$

By substituting (24) into (25) and setting $f^r_{u^m_s} = 0$, if u^m_s does not belong to ϑ_s, $s \in \mathcal{S}$, the solution of β can be obtained as

$$\beta^* = \left(\frac{1}{f}\sum_{s\in\mathcal{S}}\sum_{u^m_s\in\vartheta_s}\sqrt{\eta_{u^m_s}}\right)^2. \tag{26}$$

Substituting (26) into (24), the optimal solution can be obtained as follows:

$$\left(f^r_{u^m_s}\right)^* = \frac{f\sqrt{\eta_{u^m_s}}}{\sum_{s\in\mathcal{S}}\sum_{u^m_s\in\vartheta_s}\sqrt{\eta_{u^m_s}}}, u^m_s \in \vartheta_s, s \in \mathcal{S}. \tag{27}$$

The optimal objective function of (20) can be expressed as

$$\Phi(\mathcal{X}, \mathcal{F}^*) = \frac{\left(\sum_{s\in\mathcal{S}}\sum_{u^m_s\in\mathcal{O}_s}\sqrt{\eta_{u^m_s}}\right)}{f}. \tag{28}$$

5.3. Task Offloading Decision. In the previous section, for a given task offloading decision $\mathcal{X}$, the solutions of RBA and CRA are obtained. According to (13), (16), (19), and (28), the system utility can be expressed as follows:

$$W^*(\mathcal{X}) = \sum_{s\in\mathcal{S}}\sum_{u^m_s\in\mathcal{U}_s}\left(\beta^t_{u^m_s} + \beta^e_{u^m_s}\right) - \Gamma(\mathcal{X}, \mathcal{Y}^*) - \Phi(\mathcal{X}, \mathcal{F}^*). \tag{29}$$

Given the RB allocation strategy of $\mathcal{Y}^*$ and computing allocation strategy of $\mathcal{F}^*$, the objective function of (13) can be written as

$$\begin{aligned} &\max_{\mathcal{X}} W^*(\mathcal{X}) \\ &\text{s.t.} x_{u^m_s} \in \{0, 1\} \forall u^m_s \in \mathcal{U}_s, s \in \mathcal{S}. \end{aligned} \tag{30}$$

From the above equation, it is not a convex function due to the fact that $\mathcal{X}$ is a binary variable. For the purpose of solving this nonconvex problem, a semi-distributed TOWCRM algorithm consisting of two stages that can find a local optimum to problem (30) is adopted, as shown in Algorithm 1. In the first stage, each mobile user independently optimizes its user utility after optimizing wireless and computing resource allocation and determines whether to send an offloading request, including the information on mobile user parameters and the features of computation task. In the second stage, the MEC server determines whether the offloading user joins the offloading set by comparing the system utility, which includes the offloading user or not. Finally, the selected mobile users offload their computation tasks.

In stage 1, each UE calculates its own user utility $W_{u^m_s}$, according to $\beta^t_{u^m_s}((T^{\text{loc}}_{u^m_s} - T^r_{u^m_s})/T^{\text{loc}}_{u^m_s}) + \beta^e_{u^m_s}((E^{\text{loc}}_{u^m_s} - E^r_{u^m_s,\text{off}})/E^{\text{loc}}_{u^m_s})$. Moreover, each UE checks whether its user utility is larger than zero. If it satisfies, an offloading request is sent. Otherwise, an empty message is sent, which indicates that local computation is adopted.

In stage 2, the MEC server waits until it has collected all the requests and accepts the top N UEs of user utility in the offloading request. The initial offloading policy $\mathcal{X}$ can be got. The corresponding RB allocation strategy of $\mathcal{Y}^*$ and the corresponding computing resource allocation of $\mathcal{F}^*$ are obtained, respectively. According to (29), the system utility of $W(\mathcal{X}, \mathcal{Y}^*, \mathcal{F}^*)$ can be obtained. And let K be the set of UEs that the server accepts requests but does not accept offloading. The MEC server selects the UE with maximum user utility in K to add offloading policy $\mathcal{X}$, and the RB allocation strategy $\mathcal{Y}$ and the computing resource allocation $\mathcal{F}$ will be updated. According to (19), the system utility $W(\mathcal{X}, \mathcal{Y}, \mathcal{F})$ can be obtained. If $W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) > W(\mathcal{X}, \mathcal{Y}^*, \mathcal{F}^*)$, the MEC server removes this UE from the offloading policy. Otherwise, the system utility, RB allocation strategy, and computing resource allocation are updated. Finally, this UE is removed from the set K, and steps 21 to 33 will be repeated until the set K is equal to $\varnothing$. The MEC server forms the RB allocation strategy and computing resource allocation strategy and starts to send offloading decision to UEs. Receiving this message, UEs start to offload their tasks accordingly.

6. Simulation Results and Analysis

In a centralized MEC network, it consists of one MBS with the MEC server and four SBSs are deployed in $100 * 100$ m^2. The MBS is located in the center of the area, and the four SBSs are placed in the four directions of the area. Each SBS has a coverage area of 30 m. The radio communication parameters follow the Third Generation Partnership Project specification [33]. It is assumed that the data size of $D_{u^m_s}$ is 420 kB and the workload of $C_{u^m_s}$ is 1000 megacycles. The MATLAB® package is used to carry out the simulations, and the system parameters are summarized in Table 2.

In addition, UEs are randomly distributed in the coverage of each SBS. If not particularly indicated, the number of RBs is 10. The system utility performance of the proposed

Stage 1: at UEs side
1: for each base station s $\in \mathcal{S}$ do
2: for each device $u_s^m \in \mathcal{U}_s$ do
3: $\mathcal{Y}^* \longleftarrow$ improved graph coloring algorithm
4: $\mathcal{F}^* \longleftarrow$ equation (28)
5: $W_{u_s^m} \longleftarrow \beta_{u_s^m}^t (T_{u_s^m}^{loc} - T_{u_s^m}^r / T_{u_s^m}^{loc}) + \beta_{u_s^m}^e (E_{u_s^m}^{loc} - E_{u_s^m,off}^r / E_{u_s^m}^{loc})$
6: end for
7: end for
8: if $W_{u_s^m} > 0$ then
9: send an offloading request
10: else
11: send NULL
12: end if
Stage 2: at the MEC side
13: Wait until all requests are accepted
14: for $(i = 0; i < N; i{+}{+})$ do
15: $x_{u_s^m} \longleftarrow \arg \max (W_{u_s^m}), u_s^m \in \mathcal{U}_s, s \in \mathcal{S}$.
16: set $\mathcal{X} \longleftarrow \mathcal{X} \cup \{x_{u_s^m}\}$
17: end for
18: $\mathcal{Y}^* \longleftarrow$ step 3; $\mathcal{F}^* \longleftarrow$ step4
19: $W(\mathcal{X}, \mathcal{Y}^*, \mathcal{F}^*) \longleftarrow$ equation (29)
20: let K be the set of UEs that the server accepts requests but does not accept offloading
21: while $|K| > 0$ do
22: $x_k \longleftarrow \arg \max (W_k), k \in K$
23: $\mathcal{X} \longleftarrow \mathcal{X} \cup \{x_k\}$
24: $\mathcal{Y} \longleftarrow$ step 3; $\mathcal{F} \longleftarrow$ step 4; $W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) \longleftarrow$ step 19
25: if $W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) < W(\mathcal{X}, \mathcal{Y}^*, \mathcal{F}^*)$ then
26: $\mathcal{X} \longleftarrow \mathcal{X} / \{x_k\}$
27: else
28: $W(\mathcal{X}, \mathcal{Y}^*, \mathcal{F}^*) = W(\mathcal{X}, \mathcal{Y}, \mathcal{F})$
29: $\mathcal{Y}^* = \mathcal{Y}$
30: $\mathcal{F}^* = \mathcal{F}$
31: end if
32: $K \longleftarrow K / \{k\}$
33: end while
34: The MEC server forms RB allocation strategy $\mathcal{Y}$ and computing resources $\mathcal{F}$ and starts to send offloading decision $\mathcal{X}$ to UEs

Algorithm 1: Semi-distributed TOWCRM Algorithm.

Table 2: Basic parameters of system simulation.

Parameters	Values
RB bandwidth B	1 MHz
Number of resource blocks RB	10
UE transmitted power $P_{u_s^m}$	20 dBm
UEs preference $\beta_{u_s^m}^t = \beta_{u_s^m}^e$	0.5
Input data size $D_{u_s^m}$	420 kB
Total number of CPU cycles $C_{u_s^m}$	1000 megacycles
UE computing capacity $f_{u_s^m}^{\text{loc}}$	0.7 GHz
MEC computing capacity f	100 GHz
The background noise σ^2	-100 dBm
Pathloss from UE to SBS	$140.7 + 36.7\log10(r)$

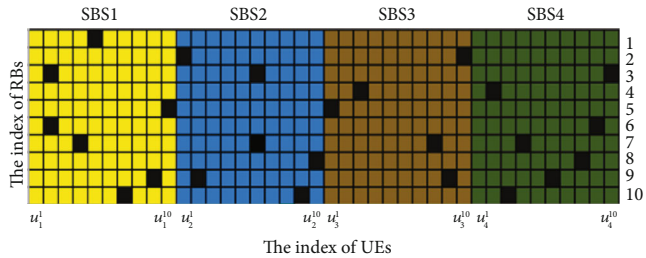
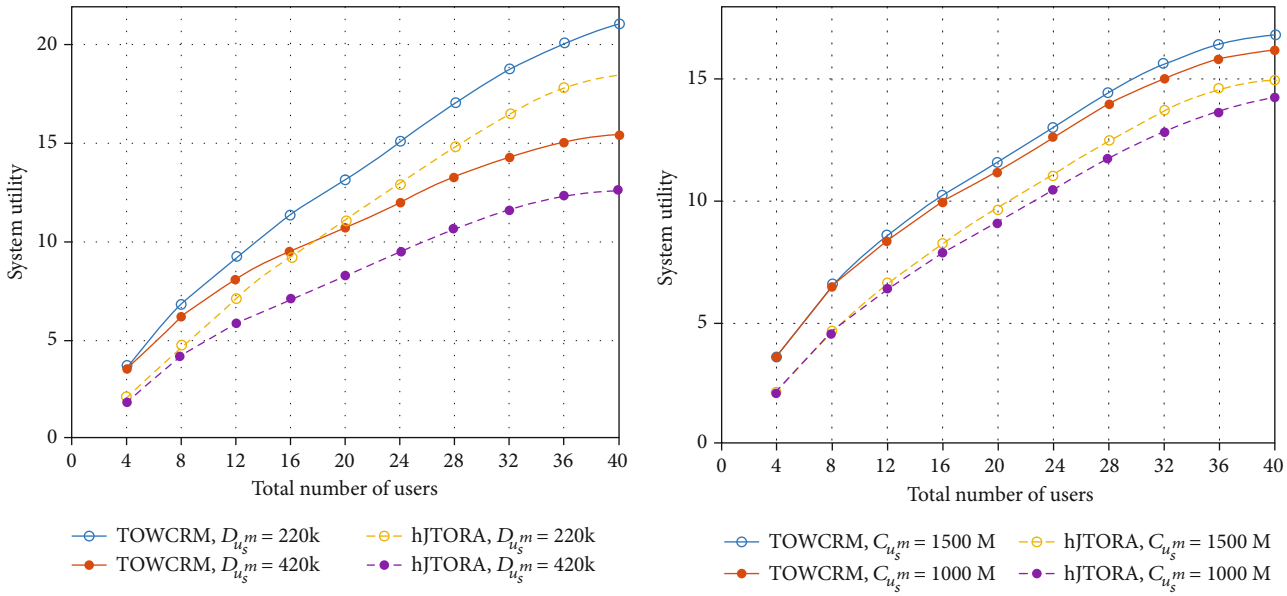


Figure 2: RB allocation based on improved graph coloring.

TOWCRM strategy is compared with that of the heuristic joint task offloading scheduling and resource allocation strategy (hJTORA) [21].

In order to visually show the resource allocation algorithm based on improved graph coloring, Figure 2 shows the RB allocation of UEs. There is a total of one MBS, four SBSs, and 10 RBs in the whole system, and there are 10 UEs associated with each SBS, among which there are 23



(a) The system utility with different input data sizes ($D_{u_s^m}$)

(b) The system utility with different workloads ($C_{u_s^m}$)

Figure 3: The system utility against different task input data sizes ($D_{u_s^m}$) or workloads ($C_{u_s^m}$).

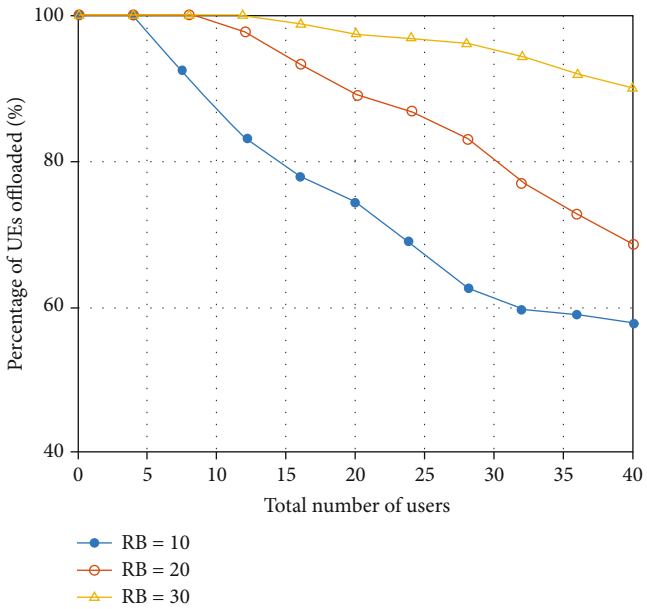


Figure 4: The relationship between the proportion of offloaded UEs and the number of UEs.

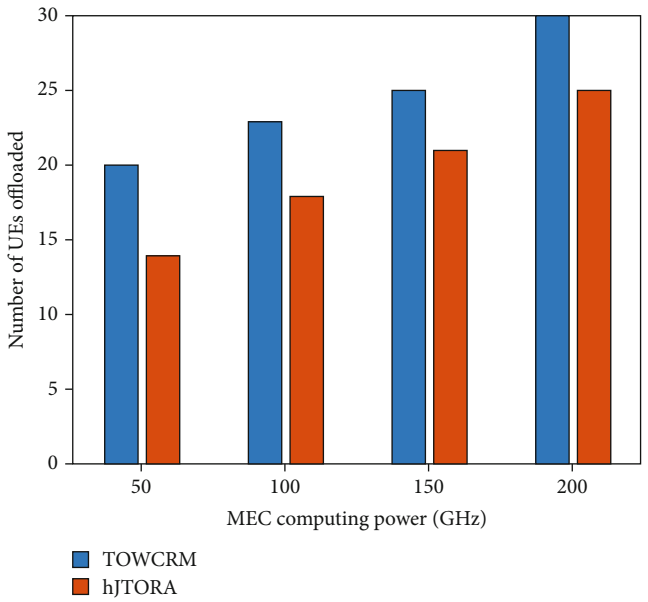


Figure 5: Comparison of the number of UEs offloaded against different MEC computing power.

offloading UEs. We can observe that the UE covered by the same SBS does not occupy the same RBs, and the RB is reused by the UEs belonging to different SBS, such as the 2-th and the 4-th RB. Some UEs are assigned to multiple RBs, such as u_1^2 and u_2^6. The results of RB allocation show that the resource matching algorithm is effective.

By performing 1000 times of simulation, Figures 3(a) and 3(b) show the system utility performance with different $D_{u_s^m}$ or $C_{u_s^m}$, respectively. From Figures 3(a) and 3(b), the system utility calculated by the proposed TOWCRM strategy is higher than that computed by hJTORA. From Figure 3(a), it can be seen that the system performance of two strategies decreases as the tasks' input data size increases. From Figure 3(b), the system utility becomes larger as the tasks' workload increases. This means that the task with smaller input data or higher workloads will improve the value of system utility.

From Figure 4, the proportion of offloading users decreases, as the number of user increases. This is mainly because the capacity of computing resources and the RBs assigned to each offloaded users decreases, as the number of users increases. Therefore, more tasks tend to be

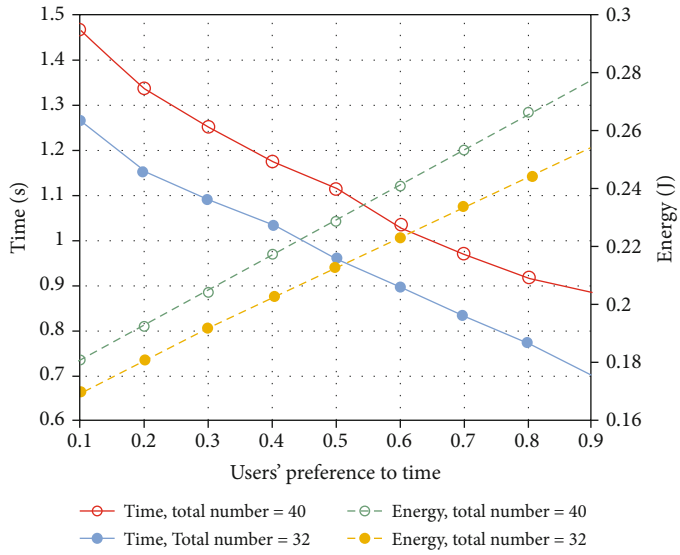


Figure 6: Time and energy consumption of all users obtained using TOWCRM.

processed locally. In addition, the proportion of offloaded users will increase with the larger number of RBs in the network.

The number of offloading UEs under different computing power is analyzed, as shown in Figure 5. It can be seen that the number of users offloaded by the TOWCRM and hJTORA algorithms is increasing with the enhancement of computing power. Because the computing power of MEC is stronger, the computation time of the offloading tasks becomes shorter. Therefore, more UEs will tend to offload their tasks to the MEC server to be processed. Moreover, under the same computing power of the MEC server, the number of UEs offloaded by the proposed algorithm is generally higher than that by using hJTORA.

Figure 6 shows the total time for finishing all offloading tasks and energy consumption obtained using TOWCRM when UEs' preferences to time of $\beta^t_{u_s^m}$ vary from 0.1 to 0.9. It can be seen that the time is reduced, and the energy consumption is increased as $\beta^t_{u_s^m}$ becomes larger. In addition, when the number of users in the system increases, the total time and energy consumption of users will be increased. This is because when more users participate in the competition for limited resources, a longer delay and higher energy consumption of all offloading UEs will occur.

7. Conclusion

In this article, the scenario of a multicell and multiuser mobile-edge computing network is modeled and analyzed. The optimization of the user utility and the system utility is formulated by combining task offloading and wireless and computing resource management. The original problem is decomposed into resource block allocation (RBA), computing resource allocation (CRA), and task offloading decision. The RBA is solved by using a resource allocation algorithm based on an improved graph coloring method. The optimal solution of CRA is obtained by using KKT conditions. In task offloading, a semi-distributed TOWCRM strategy is proposed to optimize the system utility under the constraints of computing resources. Simulation results show the effectiveness of the scheme under different system parameters. The transmission power of every user is considered a fixed value and is equal to each other for wireless resource allocation in this work. The power control of each user will be studied to improve the system utility in the next research work.

Data Availability

We derived the writing material from different journals as provided in the references. A MATLAB tool has been utilized to simulate our concept.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the 2020 State Grid Corporation of China Science and Technology Program under Grant 5700-202041398A-0-0-00.

References

[1] S. Wang, M. Zafer, and K. K. Leung, "Online placement of multi-component applications in edge computing environments," *IEEE Access*, vol. 5, pp. 2514–2533, 2017.

[2] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 870–883, 2013.

[3] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.

[4] S. Bu and F. R. Yu, "Green cognitive mobile networks with small cells for multimedia communications in the smart grid environment," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2115–2126, 2014.

[5] R. Xie, F. R. Yu, H. Ji, and Y. Li, "Energy-efficient resource allocation for heterogeneous cognitive radio networks with femtocells," *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, pp. 3910–3920, 2012.

[6] B. Yang, G. Mao, M. Ding, X. Ge, and X. Tao, "Dense small cell networks: from noise-limited to dense interference-limited," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4262–4277, 2018.

[7] X. Ge, S. Tu, G. Mao, C. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 72–79, 2016.

[8] G. Huang and J. Li, "Interference mitigation for femtocell networks via adaptive frequency reuse," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2413–2423, 2016.

[9] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing," *IEEE Access*, vol. 6, pp. 19324–19337, 2018.

[10] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via Markov decision process in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2483–2493, 2021.

[11] G. Peng, H. Wu, H. Wu, and K. Wolter, "Constrained multiobjective optimization for IoT-enabled computation offloading in collaborative edge and cloud computing," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13723–13736, 2021.

[12] C. Yi, S. Huang, and J. Cai, "Joint resource allocation for device-to-device communication assisted fog computing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 3, pp. 1076–1091, 2021.

[13] C. Guo, W. He, and G. Y. Li, "Optimal fairness-aware resource supply and demand management for mobile edge computing," *IEEE Wireless Communications Letters*, vol. 10, no. 3, pp. 678–682, 2021.

[14] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: a market equilibrium approach," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 302–317, 2021.

[15] Y. He, Y. Wang, C. Qiu, Q. Lin, J. Li, and Z. Ming, "Blockchain-based edge computing resource allocation in IoT: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2226–2237, 2021.

[16] W. Feng, H. Liu, Y. Yao, D. Cao, and M. Zhao, "Latency-aware offloading for mobile edge computing networks," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2673–2677, 2021.

[17] G. Zhang, S. Zhang, W. Zhang, Z. Shen, and L. Wang, "Joint service caching, computation offloading and resource allocation in mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 8, pp. 5288–5300, 2021.

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

[19] 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 Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, 2017.

[20] H. Zhang, L. I. Hu, S. Chen, and H. E. Xiaofan, "Computing offloading and resource optimization in ultra dense networks with mobile edge computation," *Journal of Electronics & Information Technology*, vol. 41, no. 5, 2019.

[21] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.

[22] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[23] A. Roy, S. K. Das, and A. Misra, "Exploiting information theory for adaptive mobility and resource management in future cellular networks," *Wireless Communications IEEE*, vol. 11, no. 4, pp. 59–65, 2004.

[24] L. Ma, F. Yu, V. C. M. Leung, and T. Randhawa, "A new method to support UMTS/WLAN vertical handover using SCTP," *IEEE Wireless Communications*, vol. 11, no. 4, pp. 44–51, 2004.

[25] F. Yu and V. Krishnamurthy, "Optimal joint session admission control in integrated WLAN and CDMA cellular networks with vertical handoff," *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 126–139, 2007.

[26] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[27] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.

[28] J. Liu and Q. Zhang, "Code-partitioning offloading schemes in mobile edge computing for augmented reality," *IEEE Access*, vol. 7, pp. 11222–11236, 2019.

[29] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2015.

[30] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multiuser computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[31] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming*, Springer Science & Business Media, Berlin, Germany, 2006.

[32] Y. Cheng, M. Pesavento, and A. Philipp, "Joint network optimization and downlink beamforming for CoMP transmissions using mixed integer conic programming," *IEEE Transactions on Signal Processing*, vol. 61, no. 16, pp. 3972–3987, 2013.

[33] 3rd Generation Partnership Project, "Further advancements for E-UTRA physical layer aspects," Sophia Antipolis Cedex, France, 2010, 3GPP TR 36.814, E-UTRA Access, Tech. Rep..