Research Article

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

Xintao Wu, Jie Gan, Shiyong Chen, Xu Zhao, and Yucheng Wu

1School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, China
2Beijing 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

Copyright © 2021 Xintao Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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.
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 \( x_{ms} \in \{0, 1\} \). \( x_{ms} = 0 \) means that \( u_m^s \) performs its task locally. \( x_{ms} = 1 \) means that the user of \( u_m^s \) chooses to offload the task to the MEC server via a wireless link. The task offloading decision can be expressed as \( \mathcal{X} = [x_{ms}] \), 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 \( \mathcal{N} = \{1, 2, \ldots, n, \ldots, N\} \). The RB associated table is defined as \( \mathcal{Y} = \{y_{ms}^n\} \), 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. \( y_{ms}^n = 1 \) means that the \( n \)-th RB is assigned to \( u_m^s \); otherwise, \( y_{ms}^n = 0 \). And the RB allocation strategy is defined as \( \mathcal{F} = \{Y_s\}, s \in S \).

During uplink transmission, each UE and each SBS have one antenna for sending and receiving messages. When \( u_m^s \) offloads its task to the MEC server for calculation, interference will occur if there are UE(s) in other SBSs sharing the same RB(s) with the current \( u_m^s \). As RBs are assigned orthogonally to users in each cell, there is no interference 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 \( \mathcal{U}_s = \{u_1^s, u_2^s, \ldots, u_M^s\} \), where \( u_m^s \) 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.

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 \( \mathcal{S} = \{1, 2, \ldots, s, \ldots, S\} \) be the
in intracell. The interference transmission power from \( u^m_i \) sharing the \( n \)-th RB to the \( s \)-th cell can be described as

\[
I_{u^m_i}^n = \sum_{t=1}^{M} \sum_{m=1}^{N} x_{u^n_t} y_{u^n_t} \frac{P_{u^n}}{K_{u^n}} H_{u^m_i, s},
\]

where \( P_{u^n} \) represents the transmission power of \( u^m_i \), \( K_{u^n} \) stands for the number of RBs assigned to \( u^m_i \), and \( H_{u^m_i, s} \) denotes the channel gain between \( u^m_i \) and \( s \).

Given the decision matrix \( \mathcal{X} \) and the RB associated strategy \( \mathcal{Y} \), the uploading rate achieved by \( u^n_i \) connected to \( s \) can be obtained by Shannon’s formula as [19]

\[
R_{u^n_i}(\mathcal{X}, \mathcal{Y}) = x_{u^n_i} \sum_{n=1}^{N} y_{u^n_i, n} B \log_2 \left( 1 + \frac{P_{u^n} H_{u^m_i, s}}{K_{u^n} (I_{u^n_i} + \sigma^2)} \right),
\]

where \( \sigma^2 \) is the variance of background noise, \( B \) is the bandwidth of each RB, \( P_{u^n} \) represents the transmission power of \( u^m_i \), \( K_{u^n} \) stands for the number of RB allocated to \( u^m_i \), and \( H_{u^m_i, s} \) denotes the channel gain between \( u^m_i \) and \( s \).

3.3. Calculation Model. The computing task of \( u^m_i \) is described as \( CT_{u^m_i} = C_{D_{u^m_i}, C_{u^m_i}} \), in which \( D_{u^m_i} \) (in kB) represents the size of transmission data and \( C_{u^m_i} \) (in megacycles) specifies the workload, i.e., the number of CPU cycles required to complete the computing task. The values of \( D_{u^m_i} \) and \( C_{u^m_i} \) 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 \( f_{\text{loc}} > 0 \) represent the local computing capacity of \( u^m_i \) in terms of the number of CPU cycles/s. The computation time \( T_{\text{loc}}^{\text{ct}} \) for the local execution of the task \( CT_{u^m_i} \) can be expressed as

\[
T_{\text{loc}}^{\text{ct}} = \frac{C_{u^m_i}}{f_{\text{loc}}},
\]

and the energy consumption \( E_{\text{loc}}^{\text{ct}} \) is denoted as

\[
E_{\text{loc}}^{\text{ct}} = k \left( f_{\text{loc}} \right)^2 C_{u^m_i},
\]

where \( k \left( f_{\text{loc}} \right)^2 \) 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: \( u^m_i \) 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 \( u^m_i \) can be expressed as follows:

\[
T_{u^m_i, \text{off}}^r = \frac{D_{u^m_i}}{R_{u^m_i}(\mathcal{X}, \mathcal{Y})}.
\]

When a computing task \( CT_{u^m_i} \) is offloaded to the MEC server, the MEC server allocates specific computing resources to process the task, which is represented by \( f_{u^m_i} \) (CPU cycles/s). And the computing resource allocation profile is defined as \( \mathcal{F} = \{ f_{u^m_i} \} \). 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

\[
T_{u^m_i, \text{exe}}^r = \frac{C_{u^m_i}}{f_{u^m_i}}.
\]

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

\[
\sum_{s \in \mathcal{S}} \sum_{u^m_i \in \mathcal{U}_s} x_{u^n_t, u^m_i} f_{u^m_i} \leq f.
\]

The total delay of \( u^m_i \) for finishing the task is given by the following equation:

\[
T_{u^m_i}^r = T_{u^m_i, \text{exe}}^r + T_{u^m_i, \text{off}}^r = \frac{C_{u^m_i}}{f_{u^m_i}} + \frac{D_{u^m_i}}{R_{u^m_i}(\mathcal{X}, \mathcal{Y})}.
\]

Through the above analysis, the energy consumption of \( u^m_i \) during the transmission data can be calculated as

\[
E_{u^m_i, \text{off}}^r = P_{u^m_i} \times T_{u^m_i, \text{off}}^r = P_{u^m_i} \frac{D_{u^m_i}}{R_{u^m_i}(\mathcal{X}, \mathcal{Y})},
\]

where \( P_{u^m_i} \) represents the transmitting power of \( u^m_i \).

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 \( \beta_{u^m_i} \) and energy consumption of \( \beta_{u^m_i} \), \( \beta_{u^m_i} \in [0, 1] \), and \( \beta_{u^m_i} + \beta_{u^m_i} \).
utility of computing with that of local execution \cite{18, 21}. The user utility of $W_{u^n}$ for $u^n$ can be defined as

$$W_{u^n} = \left( \frac{\beta_{u^n} T_{loc}^{u^n} - T_{u^n}^{u^n}}{T_{loc}^{u^n} + \beta_{u^n} E_{loc}^{u^n} - E_{u^n, off}^{u^n}} \right) x_{u^n}. \quad (10)$$

$\beta_{u^n}$ and $\beta_{u^n}$ 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 $W_{u^n}$ is equal to 0 when the task of $u^n$ is executed locally ($x_{u^n} = 0$). When the task of $u^n$ is executed on the MEC server ($x_{u^n} = 1$), its user utility $W_{u^n}$ is larger than 0.

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

$$W(X, Y, F) = \sum_{s \in S} \sum_{u^n \in U_s} W_{u^n}. \quad (11)$$

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

$$\max_{X, Y, F} W(X, Y, F)$$
$$\text{s.t.} \ C_1 : x_{u^n} \in \{0, 1\} \forall u^n \in U_s, s \in S$$
$$C_2 : y_{u^n} \in \{0, 1\} \forall u^n \in U_s, n \in N, s \in S,$$
$$C_3 : \sum_{s \in S} \sum_{u^n \in U_s} x_{u^n} f_{u^n}^{r} \leq f. \quad (12)$$

The constraints in the above formula can be interpreted as follows: constraint $C_1$ in (12) implies that the task can be executed locally or offloaded to the MEC server for execution. Constraint $C_2$ in (12) indicates whether the $n$-th RB is assigned to $u^n$. Constraint $C_3$ 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 \cite{31}. The equation of (12) can be rewritten as follows:

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

From (13), it can be seen that offloading decision, RB allocation, and computing resource allocation are decoupled from each other \cite{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 $O_s$.

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

$$\max_{X, F} \sum_{s \in S} \sum_{u^n \in O_s} \left( \beta_{u^n} T_{loc}^{u^n} + \beta_{u^n} E_{loc}^{u^n, off} \right) x_{u^n}$$
$$\text{s.t.} \ C_1 : y_{u^n} \in \{0, 1\} \forall u^n \in U_s, n \in N, s \in S,$$
$$C_2 : \sum_{s \in S} \sum_{u^n \in O_s} x_{u^n} f_{u^n}^{r} \leq f. \quad \text{(14)}$$

where

$$V(X, Y, F) = \sum_{s \in S} \sum_{u^n \in O_s} \left( \beta_{u^n} T_{loc}^{u^n} + \beta_{u^n} E_{loc}^{u^n, off} \right). \quad (15)$$

From (14), it is easy to see that $\sum_{s \in S} \sum_{u^n \in O_s} (\beta_{u^n} + \beta_{u^n})$ is an exact value for a specific offloading decision of $X$. The $V(X, Y, F)$ 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.

$$\min_{Y, F} V(X, Y, F) = \min_{Y, F} \left( \sum_{s \in S} \sum_{u^n \in O_s} \phi_{u^n} + \psi_{u^n} + \eta_{u^n} \right)$$
$$\text{s.t.} \ C_1 : y_{u^n} \in \{0, 1\} \forall u^n \in U_s, n \in N, s \in S,$$
$$C_2 : \sum_{s \in S} \sum_{u^n \in O_s} f_{u^n}^{r} \leq f. \quad (16)$$

where $\phi_{u^n} = \beta_{u^n} D_{u^n}/T_{loc}^{u^n}$, $\psi_{u^n} = \beta_{u^n} D_{u^n} P_{u^n}/E_{loc}^{u^n}$, and $\eta_{u^n} = \beta_{u^n} E_{loc}^{u^n, off}$.

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 $I(X, Y)$ can be written as
\[
\min_{\mathcal{Y}} \Gamma(\mathcal{X}, \mathcal{Y}) = \min_{\mathcal{Y}} \sum_{s, \delta \in \mathcal{D}} \sum_{\mathcal{F} \in \mathcal{E}} \phi_{s, \delta}^{\mathcal{F}} + \psi_{s, \delta}^{\mathcal{F}}.
\]
\[
\text{s.t. } y_{s, \delta}^{\mathcal{F}} \in \{0, 1\} \forall u_{s}^{\mathcal{F}} \in \mathcal{U}, n \in \mathcal{N}, s \in \mathcal{S}.
\]

Note that in the RB allocation phase, it is assumed that all UEs are transmitted with a fixed transmission power of \(P_{s, \delta}^{\mathcal{F}}\). The transmitted power of each UE is equally distributed over each RB assigned to it. From (17), the minimal value of the matching problem of the number of offloading UEs is obtained. 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 \(\mathcal{Y}\) to zeros and constructs \(S\) user benefit matrices as \(R_{1 \times \delta} = \{r_{s, \delta}^{\mathcal{F}}\}\), where every user rate is \(r_{s, \delta}^{\mathcal{F}} = B \log_2(1 + (P_{s, \delta}^{\mathcal{F}} H_{s, \delta}^{\mathcal{F}}) / \sigma^2)\), \(u_s^{\mathcal{F}} \in \mathcal{F}, s \in \mathcal{S}\). At the same time, each UE also constructs its own channel benefit matrix \(R_{1 \times N}^{\mathcal{F}} = \{y_{s, \delta}^{\mathcal{F}} B \log_2(1 + (P_{s, \delta}^{\mathcal{F}} H_{s, \delta}^{\mathcal{F}}) K_{s, \delta}^{\mathcal{F}} (I_{s, \delta}^{\mathcal{F}} + \sigma^2))\}\), \(u_s^{\mathcal{F}} \in \mathcal{F}, s \in \mathcal{S}\).

(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 zero frequency reuse (UZFR) method [9]. Otherwise, the elements in \(S\) user benefit matrices are sorted by \(r_{s, \delta}^{\mathcal{F}}\) 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 (step 3): 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 \(I' = \emptyset\). Then, the algorithm terminates and the RB allocation strategy \(\mathcal{Y}^*\) is obtained under the offloading decision.

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

\[
\min_{\mathcal{Y}} \Gamma(\mathcal{X}, \mathcal{Y}) = \min_{\mathcal{Y}} \sum_{s, \delta \in \mathcal{D}} \sum_{\mathcal{F} \in \mathcal{E}} \phi_{s, \delta}^{\mathcal{F}} + \psi_{s, \delta}^{\mathcal{F}}.
\]

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:

\[
\min_{\mathcal{F}} \phi(\mathcal{X}, \mathcal{F})
\]
\[
\text{s.t. } C_1 : \sum_{s, \delta \in \mathcal{D}} \sum_{\mathcal{F} \in \mathcal{E}} f_{s, \delta}^{\mathcal{F}} \leq f,
\]
\[
C_2 : f_{s, \delta}^{\mathcal{F}} > 0,
\]

where

\[
\Phi(\mathcal{X}, \mathcal{F}) = \sum_{s, \delta \in \mathcal{D}} \sum_{\mathcal{F} \in \mathcal{E}} \eta_{s, \delta}^{\mathcal{F}}.
\]

From the above equation, it is a convex optimization problem. And constraint \(C_2\) 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

\[
L \left( f_{s, \delta}^{\mathcal{F}}, \beta \right) = \sum_{s, \delta \in \mathcal{D}} \sum_{\mathcal{F} \in \mathcal{E}} \eta_{s, \delta}^{\mathcal{F}} + \beta \left( f_{s, \delta}^{\mathcal{F}} - f \right).
\]

Let \(\beta > 0\) be the Lagrange operator; the derivatives of the Lagrange function of \(L( f_{s, \delta}^{\mathcal{F}}, \beta)\) can be described as
\[
\frac{\partial L(f_{w_o}^*, \beta)}{\partial f_{w_o}^*} = -\frac{\eta_{w_o}^*}{f_{w_o}^*} + \beta.
\]

(23)

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

\[
(f_{w_o}^*)^* = \sqrt{\frac{\eta_{w_o}^*}{\beta}}.
\]

(24)

where

\[
\sum_{s \in S} \sum_{k \in D} (f_{w_o}^*)^* = f.
\]

(25)

By substituting (24) into (25) and setting \(f_{w_o}^* = 0\), if \(u_{w_o}^*\) does not belong to \(\vartheta_1, s \in \vartheta\), the solution of \(\beta^*\) can be obtained as

\[
\beta^* = \left(\frac{1}{f} \sum_{s \in \vartheta} \sum_{k \in D} \sqrt{\eta_{w_o}^*}\right)^2.
\]

(26)

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

\[
(f_{w_o}^*)^* = \frac{f \sum_{s \in \vartheta} \sqrt{\eta_{w_o}^*}}{\sum_{s \in \vartheta} \sum_{k \in D} \sqrt{\eta_{w_o}^*}}, u_{w_o}^* \in \vartheta_1, s \in \vartheta.
\]

(27)

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

\[
\Phi(\mathcal{X}, \mathcal{F}^*) = \left(\frac{\sum_{s \in \vartheta} \sum_{k \in D} \sqrt{\eta_{w_o}^*}}{f}\right).
\]

(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:

\[
W^*(\mathcal{X}) = \sum_{s \in \vartheta} \sum_{k \in D} \left(\beta_{w_o}^* + \beta_{e_o}^*\right) - \Gamma(\mathcal{X}, \mathcal{Y}^*) - \Phi(\mathcal{X}, \mathcal{F}^*).
\]

(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

\[
\max_\mathcal{X} W^*(\mathcal{X})
\]

subject to

\[
\text{s.t.} X_{w_o}^* \in \{0, 1\} \forall w_o^* \in \mathcal{U}_s, s \in \vartheta.
\]

(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 \(W_{w_o}^*\), according to \(\beta_{w_o}^* (\gamma_{w_o}^* + (\ell_{loc} - T_{w_o}^*) / T_{w_o}^*) + \beta_{e_o}^* (\ell_{loc} - E_{w_o}^* - E_{w_o}^*_{e_{off}})/ E_{w_o}^*\), where \(E_{w_o}^*\) is a binary variable. For the purpose of determining whether the community 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 to the system utility, and the RB allocation strategy \(\mathcal{Y}\) and the computing resource allocation \(\mathcal{F}\) will be updated. According to (19), the system utility of \(W(\mathcal{X}, \mathcal{Y}, \mathcal{F})\) 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 m². 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 m². The radio communication parameters follow the Third Generation Partnership Project specification [33]. It is assumed that the data size of \(D_{w_o}^*\) is 420 kB and the workload of \(C_{w_o}^*\) 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 area of each SBS. If not particularly indicated, the number of RBs is 10. The system utility performance of the proposed
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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>RB bandwidth B</td>
<td>1 MHz</td>
</tr>
<tr>
<td>Number of resource blocks RB</td>
<td>10</td>
</tr>
<tr>
<td>UE transmitted power $P_{ums}$</td>
<td>20 dBm</td>
</tr>
<tr>
<td>UEs preference $\beta^{e}<em>{ums} = \beta^{t}</em>{ums}$</td>
<td>0.5</td>
</tr>
<tr>
<td>Input data size $D_{ums}$</td>
<td>420 kB</td>
</tr>
<tr>
<td>Total number of CPU cycles $C_{ums}$</td>
<td>1000 megacycles</td>
</tr>
<tr>
<td>UE computing capacity $f^{loc}_{ums}$</td>
<td>0.7 GHz</td>
</tr>
<tr>
<td>MEC computing capacity $f$</td>
<td>100 GHz</td>
</tr>
<tr>
<td>The background noise $\sigma^2$</td>
<td>-100 dBm</td>
</tr>
<tr>
<td>Pathloss from UE to SBS $140.7 + 36.7\log_{10}(r)$</td>
<td></td>
</tr>
</tbody>
</table>
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 $u_2$ and $u_6$. 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 $D_{um}$ or $C_{um}$, 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
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 h|TORA 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 h|TORA.

Figure 6 shows the total time for finishing all offloading tasks and energy consumption obtained using TOWCRM when UEs’ preferences to time of $\beta_{i,n}$ vary from 0.1 to 0.9. It can be seen that the time is reduced, and the energy consumption is increased as $\beta_{i,n}$ 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


