

## Research Article

# An Energy-Aware Task Offloading Mechanism in Multiuser Mobile-Edge Cloud Computing

Lan Li, Xiaoyong Zhang , Kaiyang Liu , Fu Jiang , and Jun Peng 

*School of Information Science and Engineering, China Hunan Engineering Laboratory of Rail Vehicles Braking Technology, Central South University, Changsha, Hunan 410083, China*

Correspondence should be addressed to Xiaoyong Zhang; zhangxy@csu.edu.cn

Received 31 March 2017; Revised 9 November 2017; Accepted 31 December 2017; Published 2 April 2018

Academic Editor: Bartolomeo Montrucchio

Copyright © 2018 Lan Li 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 cloud computing, an emerging and prospective computing paradigm, can facilitate the complex application execution on resource-constrained mobile devices by offloading computation-intensive tasks to the mobile-edge cloud server, which is usually deployed in close proximity to the wireless access point. However, in the multichannel wireless interference environment, the competition of mobile users for communication resources is not conducive to the energy efficiency of task offloading. Therefore, how to make the offloading decision for each mobile user and select its suitable channel become critical issues. In this paper, the problem of the offloading decision is formulated as a 0-1 nonlinear integer programming problem under the constraints of channel interference threshold and the time deadline. Through the classification and priority determination for the mobile devices, a reverse auction-based offloading method is proposed to solve this optimization problem for energy efficiency improvement. The proposed algorithm not only achieves the task offloading decision but also gives the facility of resource allocation. In the energy efficiency performance aspects, simulation results show the superiority of the proposed scheme.

## 1. Introduction

Internet of things (IoT) devices, such as sensors and wearable devices, are increasingly penetrating into our everyday lives. Gartner forecasted that, by 2020, the Internet of things (IoT) devices will reach 50 billion, representing an almost 30-fold increase from 0.9 billion in 2009 [1]. However, it is well known that mobile devices have their inherent problems, such as finite computing power and particularly limited battery life [2]. These resource-limited mobile devices are difficult to support computation-intensive applications, such as interactive gaming, image/video processing, and online social networking services [3]. Therefore, when dealing with the sophisticated applications on devices, the contradiction between the requirements of network applications and the limited resource of mobile devices poses a significant challenge.

A new architecture and technology known as Mobile Cloud Computing (MCC) brings a new idea, which can augment the processing ability of mobile devices and reduce the energy consumption of mobile devices at the

meantime by migrating computational tasks from mobile devices to infrastructure-based cloud servers [4, 5]. Thus, MCC has the potential to address the aforementioned challenge [6].

In recent years, cloud offloading technologies have been widely studied by researchers all over the world. Considering the huge increase of computational demand on the mobile devices in the 5G networking, Chen [7] proposed a game theoretic approach for the distributed task offloading decision to improve the energy efficiency. Barbarossa et al. [8] investigated the problem of multiuser computation offloading in multiradio channel scenarios ignoring the communication interference. In order to save the energy consumption of mobile devices and meet the requirements of the application execution time, Huang et al. [9] proposed a Lyapunov optimization-based algorithm for dynamic offloading, which can reduce the computational complexity at the same time to obtain a near-optimal offloading decision. In large-scale mobile applications, Yang et al. [10] researched the problem of multiuser computing partition for the purpose of minimizing the average task completion of all

users and designed an offline heuristic offloading algorithm. Viswanathan et al. [11] proposed a resource provisioning framework for organizing the heterogeneous devices in the vicinity. A joint optimization framework of the wireless resource and computing resource is proposed in [12] for the energy-constrained mobile users in the femtocell.

In order to solve the problem of transmission delay and energy consumption, Satyanarayanan et al. [13] proposed an architecture replacing the remote cloud with nearby cloudlets. Zhang et al. [14] proposed a Markov decision process offloading algorithm for mobile users in an intermittently connected cloudlet system. However, the cloudlet-based mobile cloud computing has some drawbacks. Due to the limited wireless network coverage, the cloudlet cannot guarantee ubiquitous service everywhere for users. Moreover, the computation resource of the cloudlet is insufficient to satisfy the QoS requirements of a large number of users in the future.

These existing task offloading strategies in mobile cloud computing are not sufficient to greatly improve the energy efficiency of the system. The long transmission delay between the mobile device and traditional cloud servers is a critical issue, which is the inherent limitation of mobile cloud computing [15, 16]. The long propagation distance from the mobile device to the remote cloud data center will result in unacceptable long latency for mobile applications. The additional communication transmission delay will decrease the computation offloading efficiency and the QoS of users.

Therefore, a mobile-edge cloud (MEC) computing architecture is adopted. With the development of wireless communication technologies such as Wi-Fi, 4G, and 5G, the MEC is envisioned as a promising and challenging approach to address the abovementioned challenges [17]. In the MEC framework, mobile devices are able to offload their tasks to the MEC clouds through the radio access points nearby rather than the public clouds such as Amazon EC2 and Windows Azure. Thus, this MEC paradigm can provide lower latency and high communication rate and computing agility in the process of computation offloading.

There are a few studies on the efficient computation offloading mechanism of the MEC. For instance, Wang et al. [18] used the Markov decision process to formulate a sequential offloading decision-making problem for dynamic service migration. Considering the finite number of wireless access channels and the interference, Chen et al. [19] presented a distributed offloading decision method based on game theory. Sardellitti et al. [20] investigated the task offloading problem by jointly considering the allocation of radio resources and computational resources and proposed an iterative algorithm to solve the problem. Beck et al. [21] studied the virtual network embedding problems and proposed network virtualization in the context of MEC networks.

However, the abovementioned methods still have some limitations in performance and flexibility. Furthermore, the complexity of the algorithm is not suitable for large-scale network scenarios. Thus, an efficient task offloading mechanism is designed for mobile-edge cloud computing. It is well known that the base stations in most wireless networks are

running under multichannel settings. If a considerably large number of mobile device users simultaneously choose the same wireless channel to perform task offloading, interactive communication interference will seriously affect the transmission rate of the data, which further leads to the increase of completion delay of the task and the energy consumption of mobile devices. In this circumstance, the task offloading operation violates the original intention of the mobile users, and these mobile users prefer local processing. The abovementioned effect makes the offloading decision of mobile device users couple with the wireless resource allocation. Thus, to achieve efficient task offloading, two important issues need to be solved: Which tasks are suitable for offloading while others are suitable for local execution? How to choose the appropriate channel performing offloading from a global optimization?

In order to deal with the abovementioned coupling problems and match the mobility of the mobile device, an energy-aware task offloading mechanism is designed to solve the energy-efficient task offloading decision problem for mobile-edge cloud computing in the multichannel wireless environment. The mechanism adopts the auction theory, which is a decentralized market mechanism of resource allocation in economy, which has been widely used in various fields nowadays, such as the management of the spectrum resource in cognitive networks [22, 23] and traffic offloading in the cellular network [24, 25]. Inspired by the methods for resource allocation, the auction is applied into the model of wireless channel access opportunities for mobile devices, namely, task offloading decision. In the auction system, the mobile device user acts as a buyer, and the wireless channels are treated as sellers. Thus, the mobile device broadcasts the offloading request, and wireless channels will send their bid information with their available resources to the mobile device user. Then, the mobile device calculates the cost to determine the winner channel to offload.

The contributions of this paper are listed as follows:

- (i) The issue of the task offloading decision is formulated as a constrained 0-1 nonlinear integer programming problem, which minimizes the total energy consumption of the system subjected to the latency and channel communication quality constrains.
- (ii) To reduce the complexity of solving the optimization problem, an algorithm at first for classifying mobile devices and priority determination is proposed. The mobile devices are classified into two groups. First category is suitable for local processing, and the remaining mobile devices are classified as the second category, which is determined by the corresponding priority.
- (iii) In order to further determine the offloading decision for the mobile device, an auction-based approach is proposed to solve the energy-efficient task offloading decision problem, with high efficiency to approximate the optimal task offloading decisions. Hence, each mobile device can minimize its average energy consumption.

The rest of this paper is organized as follows. In Section 2, the system model is described and the issue of task offloading decision is formulated as a 0-1 nonlinear integer programming problem. The details for solving the constrained optimization problem and the proposed energy-aware task offloading mechanism are provided in Section 3. In Section 4, the simulation results and the analysis of the results are presented. Finally, concluding remarks are drawn in Section 5.

## 2. System Model and Problem Formulation

In this section, the mobile-edge cloud computing architecture is presented, and the 0-1 nonlinear integer programming problem is formulated for energy saving and delay decreasing. Table 1 lists all the important symbols used in this paper.

**2.1. Mobile-Edge Cloud Computing Architecture.** As illustrated in Figure 1, the mobile-edge cloud computing system consists of three key components: mobile-edge cloud server, wireless access point, and device users. Mobile device users first connect to Internet through a base station, which is equipped with a mobile-edge cloud server. Then, the users can further access this powerful mobile-edge cloud server through the Internet. After receiving the service request, the cloud control provides the corresponding cloud services to the mobile devices via the base station. In this paper, a set of  $\mathcal{N} = \{1, 2, \dots, N\}$  devices are considered, where each device has a task to be completed. Moreover, the wireless base station runs  $\mathcal{M} = \{1, 2, \dots, M\}$  channel settings.

The types of tasks include interactive gaming, natural language processing, virus scanning, and video transcoding. Each task is set in two terms as  $\mathcal{T}_i = \Delta(O_i, D_i)$ ,  $i \in \mathcal{N} = \{1, 2, \dots, N\}$ , which can be computed either locally on the mobile device or on the cloud side via computation offloading. It is assumed that, in the model, each task is atomic and cannot be further divided. The characteristic information of all tasks can be defined through the total number of  $\{O_i\}$  CPU cycles required to accomplish and the amount of  $\{D_i\}$  data to be exchanged. The computation overhead is discussed in terms of both energy consumption and processing time for both local and cloud computing approaches.

For the local computing approach, a mobile user  $i$  executes the task  $\mathcal{T}_i$  locally on the mobile device. Taking into account the different processing capabilities of different mobile devices, as well as a different CPU computing capacity per bit required for different tasks, let  $C_i^l$  be the computation capability (i.e., CPU cycles per second) allocated by the mobile device  $i$ . Then, the time duration of the local execution of the task  $\mathcal{T}_i$  can be obtained as follows:

$$t_i^l = \frac{O_i}{C_i^l}. \quad (1)$$

The classic CPU energy consumption model of mobile devices is applied [26, 27]. Thus, the energy consumption  $E_i^l$  of this local execution can be calculated as

TABLE 1: Important notations.

Symbol	Definition
$\mathcal{N}$	Mobile IoT device set
$\mathcal{M}$	Wireless channel set
$\mathcal{T}_i$	The information of task on the device $i$
$O_i$	The computing workload of the task $i$
$D_i$	The data size for communication of the task $i$
$C_i^l$	The computing capacity required for the task $i$
$C_i^c$	The computation capacity allocated from the cloud
$P_i^{\text{tr}}$	The data transmission power of the device $i$
$P_i^{\text{id}}$	The idle power of the device $i$
$G_i$	The channel gain
$\psi_{i,j}$	The offloading decision
$w_j$	The bandwidth of the channel $j$
$R_{i,j}$	The transmission rate of the mobile device $i$
$t_i^l$	Local completion time of the task $i$
$t_i^c$	The cloud processing delay of the task $i$
$E_i^l$	Local processing energy consumption
$E_{i,j}^{\text{tr}}$	The data transmission power consumption of the task $i$
$E_i^{\text{id}}$	The idle power consumption of the mobile device $i$
$E_{i,j}^{\text{off}}$	The total offloading energy consumption of the mobile device

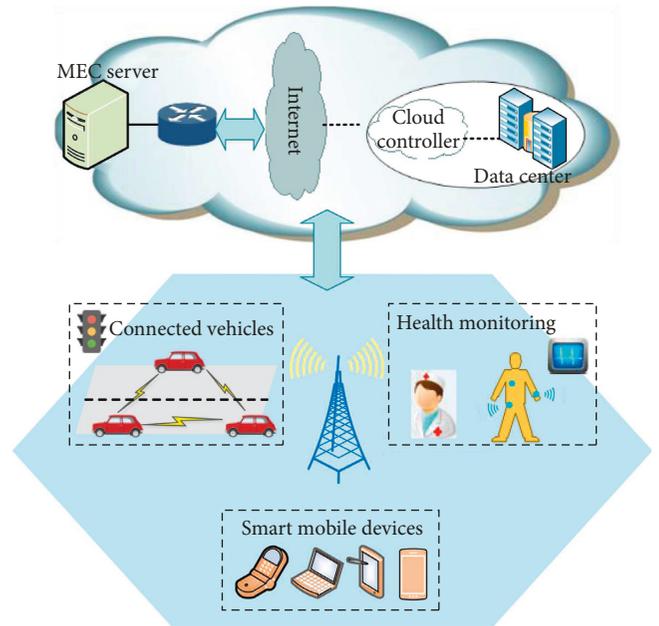


FIGURE 1: The architecture of mobile-edge cloud computing.

$$E_i^l = \left( \alpha_i (C_i^l)^{\chi_i} + \beta_i \right) t_i^l, \quad (2)$$

where the exponents  $\chi_i$ ,  $\alpha_i$ , and  $\beta_i$  are the parameters, which are dependent on the CPU processing model. All the energy consumption parameters are set to conform to the actual mobile scenario in the subsequent simulation experiment.

For the cloud computing approach, once the communication link is established, the task  $\mathcal{T}_i$  on the mobile

device can be offloaded through the channel  $j$ ,  $j \in \mathcal{M}$ , to the mobile-edge cloud server, which will execute the computation task on behalf of the device user. The whole task offloading process can be divided into three phrases and involves corresponding time delay and energy consumption.

To begin with, the mobile device transmits the data load of the task  $i$  to the closest base station through the channel  $j$ , which involves the data transmission delay  $t_{i,j}^{\text{tr}}$  and energy consumption of the data transmission  $E_{i,j}^{\text{tr}}$  of the task. The energy consumption of the data transmission is not only related to the data size but also to the uplink data rate. Thus, the binary variable  $\psi_{i,j} \in \{0, 1\}$  is denoted as the task offloading decision of the mobile user  $i$ , which is given by

$$\psi_{i,j} = \begin{cases} 1, & \text{offload the task of device } i \text{ via channel } j, \\ 0, & \text{process task of device } i \text{ locally,} \end{cases} \quad (3)$$

where  $\psi_{i,j} = 1$  means that the device  $i$  chooses to offload the task and transmits the computation data to the cloud through the channel  $j$ ,  $j \in \mathcal{M}$ , while  $\psi_{i,j} = 0$  denotes that the device executes the task locally. According to the offloading decisions, an appropriate number of VMs are deployed in the data servers for cloud execution. Moreover, given the offloading decisions  $\Psi = (\psi_{1,j}, \psi_{2,j}, \dots, \psi_{N,j})$  of all device users, the uplink data rate  $R_{i,j}$  of a device user  $i$  who chooses to offload the task to the cloud via the wireless channel  $j$  can be calculated as [19]

$$R_{i,j}(\Psi) = w_j \log_2 \left( 1 + \frac{P_i^{\text{tr}} G_i}{\sigma^2 + \sum_{r \in \mathcal{N} \setminus \{i\}; \psi_{i,j}=1} P_r^{\text{tr}} G_r} \right), \quad (4)$$

where  $w_j$  is the  $j$  channel bandwidth and  $P_i^{\text{tr}}$  represents the data transmission power of mobile devices, which is determined by the wireless base station according to some power control algorithms such as [28] and [29]. Furthermore,  $G_i = \ell_i^{-a}$  denotes the channel gain between the device user  $i$  and the base station, where  $\ell_i$  indicates the distance between the mobile device  $i$  and the wireless base station. Moreover,  $\sigma^2$  denotes the background noise power, and the parameter  $a$  denotes the path loss factor. Therefore, the data transmission delay  $t_{i,j}^{\text{tr}}$  can be calculated as

$$t_{i,j}^{\text{tr}} = \frac{D_i}{R_{i,j}}. \quad (5)$$

Let  $P_i^{\text{tr}}$  be the data transmission power consumption. The energy consumption of the data transmission can be denoted as

$$E_{i,j}^{\text{tr}} = \frac{D_i}{R_{i,j}} P_i^{\text{tr}}. \quad (6)$$

In the second phase, the base station transmits the data load to the mobile-edge cloud server through a high-speed wire network [27, 28]. Because of the high-speed link, the time delay of this phase can be ignored. In the final phase,

the mobile-edge cloud server processes the task and returns the results back to the device users. Because the size of results is often considerably smaller than that of the input data load, the time delay from the mobile-edge cloud server to the device is not considered, as in some of the previous studies. Therefore, the delay of this phase is mainly composed of the cloud processing delay  $t_i^c$ . Let  $C_i^c$  be the computation capacity allocated from the cloud. The cloud processing time  $t_i^c$  spent on the cloud side can be formulated by

$$t_i^c = \frac{O_i}{C_i^c}. \quad (7)$$

When the task is executed on the cloud, the mobile device needs to wait for the return of the response result. Thus, at the period of time  $t_i^c$  spent on the task processing on the cloud side, the idle power consumption of the mobile device can be calculated as follows:

$$E_i^{\text{id}} = \frac{O_i P_i^{\text{id}}}{C_i^c}. \quad (8)$$

Thus, the total offloading energy consumption  $E_{i,j}^{\text{Off}}$  is expressed as  $E_{i,j}^{\text{Off}} = E_{i,j}^{\text{tr}} + E_i^{\text{id}}$ , which is defined as the sum of the energy spent to transmit data to the cloud,  $E_{i,j}^{\text{tr}}$ , plus the idle power consumption,  $E_i^{\text{id}}$ , as follows:

$$E_{i,j}^{\text{Off}} = \frac{D_i P_i^{\text{tr}}}{R_{i,j}} + \frac{O_i P_i^{\text{id}}}{C_i^c}. \quad (9)$$

And the total time delay of offloading is denoted as  $T_{i,j}^{\text{Off}} = t_i^c + t_{i,j}^{\text{tr}}$ , which is defined as the transmission delay plus the cloud processing delay.

In order to realize energy-efficient task offloading, it is necessary to properly deal with the reasonable allocation of communication resources and computing resources, which are mutually coupled in the case of energy efficiency because of the competition for the resource. However, in this paper, it is assumed that the processing capacity of cloud services is far greater than the processing capacity of each mobile device, and the computing and storage resources are sufficient to satisfy the requirements of all mobile devices. Besides, our research scope of this paper is mainly focused on the mobile devices for the purpose of saving energy by offloading the task onto the cloud side. Therefore, the server overhead and energy consumption of the cloud server are not considered, which does not affect the completeness of the paper.

Among the multiple mobile device users for the mobile-edge cloud computing environment, the mobile device selects the nearest wireless access point in order to get better communication and interaction. Similar to the previous research on mobile-edge cloud computing, from the beginning of the offloading decision until the end of offloading operation, it is reasonable to assume that all mobile devices move very slowly in a quasi-static scenario.

To minimize the total energy consumption of the system, the optimization problem is mathematically modeled as follows:

$$\min_{\{\psi_{i,j}\}} \mathcal{F} = \sum_{i=1}^{\mathcal{N}} \left( (1 - \psi_{i,j}) E_i^l + \psi_{i,j} E_{i,j}^{\text{Off}} \right), \quad (10)$$

$$\text{s.t.} \quad \sum_{j=1}^{\mathcal{M}} \psi_{i,j} E_{i,j}^{\text{Off}} \leq E_i^l, \forall i \in \mathcal{N}, \quad (11)$$

$$\sum_{j=1}^{\mathcal{M}} \psi_{i,j} T_{i,j}^{\text{Off}} \leq T_i^l, \forall i \in \mathcal{N}, \quad (12)$$

$$\sum_{i=1}^{\mathcal{N}} \psi_{i,j} P_i^{\text{tr}} G_i \leq \mathcal{C}, \forall j \in \mathcal{M}, \quad (13)$$

$$\sum_{j=1}^{\mathcal{M}} \psi_{i,j} \leq 1, \forall i \in \mathcal{N}, \quad (14)$$

$$\psi_{i,j} \in \{0, 1\}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M}. \quad (15)$$

Constraint (11) ensures that the energy consumption of task offloading is not greater than the local processing energy consumption of the mobile device. Constraint (12) ensures that the total time consumption of mobile devices in the process of task offloading is not greater than the local processing energy consumption of the task. Constraint (13) is to guarantee the communication quality of the wireless channel. The setting of the threshold  $\mathcal{C}$  can avoid mobile devices to access the same channel at the same time because the burst data traffic of mobile devices will seriously cause the attenuation of channel quality. Constraint (14) states that the mobile device can only select access to a wireless channel, but the wireless channel can be accepted by a plurality of mobile devices. Constraint (15) states that the cloud offloading decision of the task is a binary variable.

Consider that the task offloading decisions  $\Psi$  among the device users are coupled. If too many device users simultaneously choose to offload the task to the cloud via the same wireless channel, they may cause severe interference, which will lead to a low data rate. The two factors related to the energy consumption of data transmission of the mobile device are the inherent transmission power and data transmission time. The transmission energy consumption of mobile devices is proportional to the transmission time. Thus, when the data rate of the mobile device user is low, it would consume high energy and incur long transmission time as well. In this case, more and more device users will avoid offloading and are more willing to choose execution locally. However, this is not our original intention: allowing beneficial cloud computing users to offloading as much as possible. Thus, the  $\mathcal{C}$  threshold is set, which can be flexible assignment.

### 3. Energy-Aware Task Offloading Mechanism

To solve the optimization problem (10), an energy-aware task offloading mechanism is designed in the system of mobile-edge cloud computing. The proposed mechanism mainly includes two aspects:

- (1) At the beginning, an algorithm for mobile device user classification and priority determination is designed. The mobile device users can be classified into two types: participation in the auction and not

to participate according to the energy cost features of the task computing process. Namely, the mobile device users who do not participate in the auction choose to process the task locally. Then, the priorities of the first class of users are derived, which represent the intensity of user demand for task offloading.

- (2) According to the order of priority, the device users get resource allocation in turn. A reverse auction-based offloading algorithm is proposed to achieve the offloading decision and associate the suitable communication resource with each mobile device who participates in the auction.

**3.1. Mobile Device User Classification and Priority Determination.** Based on the characteristics of the task and the mobile device, such as the data size of the task, workload density, computing capacity, and energy consumption, the mobile device users are divided into two types.

The first type of users is a group that should compute their task locally. The set of users of this type is denoted as  $\mathcal{S}_l$ . When the mobile device occupies a channel alone, the data transmission rate of this mobile device can be expressed by

$$R_{i,j}^0 = w_j \log_2 \left( 1 + \frac{P_i^{\text{tr}} G_i}{\sigma^2} \right). \quad (16)$$

The condition used to determine the devices belonging to this type is given as follows:

**Theorem 1.** *if  $E_i^l < E_{i,j}^{\text{Off}}$ ,  $i \in \mathcal{N}$ ,  $j \in \mathcal{M}$ , then the device  $i$  belongs to  $\mathcal{S}_l$ , where*

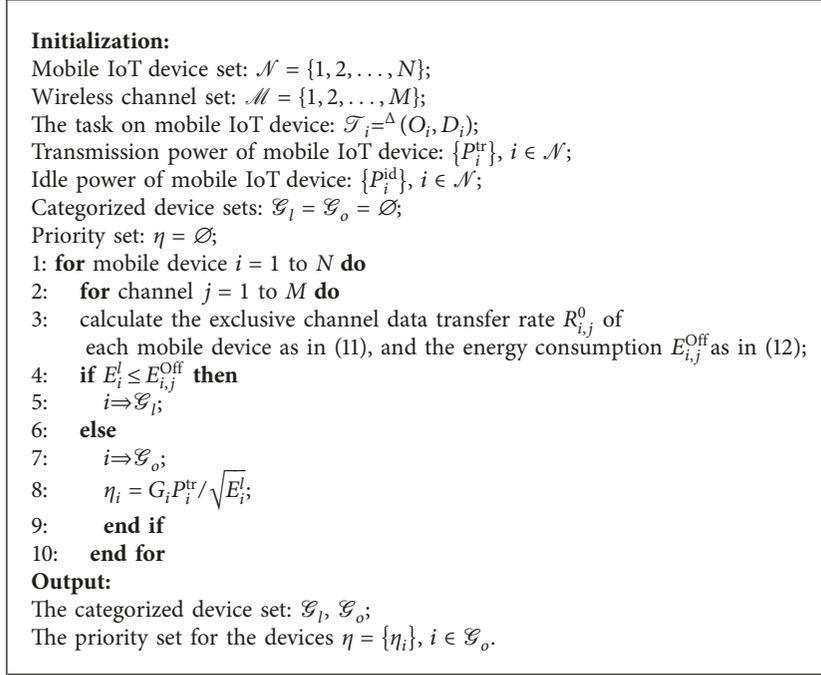
$$E_{i,j}^{\text{Off}} = \frac{D_i P_i^{\text{tr}}}{R_{i,j}^0} + \frac{O_i P_i^{\text{pid}}}{C_i^c}. \quad (17)$$

Besides the aforementioned type, the rest of the mobile device users fall into second type  $\mathcal{S}_o$ . The mobile device users belonging to  $\mathcal{S}_o$  can either decide to implement their task locally or to offload the task onto the mobile-edge cloud server. The decision of them depends on the communication quality of the channel. For this type of mobile users, different priorities are set for them in the offloading process, which is defined as

$$\eta_i = \frac{G_i P_i^{\text{tr}}}{\sqrt{E_i^l}}. \quad (18)$$

The complete mobile device user classification and priority determination are illustrated in Algorithm 1.

**3.2. Reverse Auction-Based Offloading Algorithm.** In this section, a reverse auction-based offloading scheme is proposed for the group  $\mathcal{S}_o$  of mobile devices based on the abovementioned analysis of mobile device user classification and priority determination. Our aim is to maximize the energy efficiency of task offloading subjected to the mobile device's minimum energy consumption requirement and



ALGORITHM 1: The Algorithm for classifying the mobile device and priority determination.

the limited communication resource of channels during the task offloading process.

As illustrated in Figure 2, the mobile device  $i, i \in \mathcal{G}_o$ , acts as the buyer, who achieves higher system energy efficiency in exchange of transmission power resources provided by the channel. Prior to participation in the auction, namely, deciding whether to offload the task onto the mobile-edge cloud server, mobile users first calculate their cost  $\mathcal{P}_i$ , which means the reserved prices the mobile device can accept. In this case, the reserve price corresponds to the aforementioned local computing energy consumption of the task in the system of mobile-edge cloud computing. Thus, the reserve price  $\mathcal{P}_i$  is expressed as  $\mathcal{P}_i = E_i^l$ .

On the other hand, the wireless channels are sellers. Each channel can participate in the auction by submitting to the mobile device the bidding information  $(b_j, s_j)$ , where  $b_j$  is the price at which the  $j$ th channel agrees to share their available resources to the mobile device. Each seller calculates their  $s_j$  and  $b_j$ , respectively, by  $s_j = P_i^{\text{tr}} G_{i,j}$  and  $b_j = E_{i,j}^{\text{off}}$ . The total number of available resources of each seller corresponds to the aforementioned interference threshold of each channel. Then, the mobile device will calculate the energy cost and compare the bidding prices provided by the seller to decide whether it could achieve energy saving and choose the target channel or give up task offloading decision. The target of the wireless channel is the winner of the reverse auction process. The mobile device chooses the target channel for task offloading.

In the previous auction researches, they allocate the resources through multiround bidding procedures to determine the final winner. However, this multiround auction method is not suitable for our scenario because mobile device users have to wait for the consequences after multiple rounds of auction, which inevitably generate an intolerable

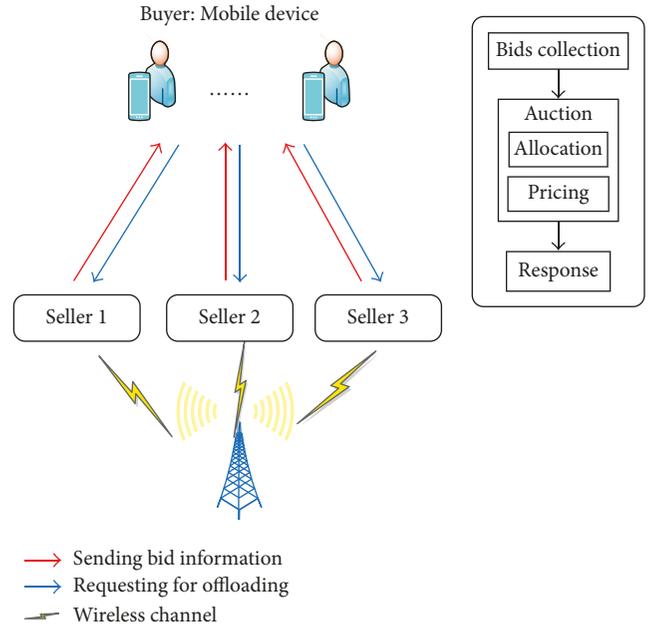


FIGURE 2: The reverse auction system. The mobile device sends the request of transmission with the energy cost, collects the bid information sent by wireless channels, and then chooses the winner channel.

extra delay. In the process of task offloading, mobile devices are sensitive to delay. Therefore, the single-round auction is implemented in this paper in order to improve the energy efficiency and reduce the delay for the offloading users. Moreover, it is assumed that the time delay of the auction process is so small that it can be ignored. The auction is conducted periodically, which means that, after a smaller

**Input:**  $\mathcal{E}_1, \mathcal{E}_o, \eta$ .  
**Output:** Offloading decision  $\Psi = (\psi_{1,j}, \psi_{2,j}, \dots, \psi_{N,j})$ .  
1: Set the temporary set  $\mathcal{E}'_o = \mathcal{E}_o$ ;  
2: **while**  $\mathcal{E}'_o \neq \emptyset$  **do**  
3:   Select the device  $i$ , where  $i = \operatorname{argmax}\{\eta_i\}_i, i \in \mathcal{E}_o$ ;  
4:   **for** channel  $j = 1$  to  $M$  **do**  
5:     Update the data transmission rate  $R_{i,j}$  and update  $E_{i,j}^{\text{Off}}$ , as in (4) and (9);  
6:     **if**  $\mathcal{E}_j > 0$  **then**  
7:       Calculate the bid density  $bd_j$  of each channel  $j$  based on the 2-tuple  $(b_j, s_j)$ ;  
8:       Set the bid density  $bd = \{bd_j\}$ ;  
9:       **while**  $bd \neq \emptyset$  **do**  
10:          Select the channel  $j$ , where  $j = \operatorname{argmin}\{bd_j\}_j$ ;  
11:          **if**  $E_{i,j}^{\text{Off}} \leq E_i$  &  $\mathcal{E}_j - \sum_{r=1, r \neq i}^{\mathcal{N}} \psi_{r,j} P_i^{\text{tr}} G_i > 0$  **then**  
12:           Let  $\psi_{i,j} = 1$ ;  
13:            $\mathcal{E}_j \leftarrow \mathcal{E}_j - \sum_{r=1, r \neq i}^{\mathcal{N}} \psi_{i,j} P_i^{\text{tr}} G_i$ ;  
14:           **else**  
15:           Let  $\psi_{i,j} = 0$ ;  
16:           **end if**  
17:            $bd = bd \setminus j$ ;  
18:          **end while**  
19:       **else**  
20:       Let  $\psi_{i,j} = 0$ ;  
21:       **end if**  
22:     **end for**  
23:      $\mathcal{E}'_o = \mathcal{E}'_o \setminus i$   
24: **end while**

ALGORITHM 2: Reverse Auction-Based Offloading Algorithm for Offloading Decisions.

auction interval, a new round of auction is started and the relevant information is collected again, which is adapted to the dynamic mobile cloud computing environment. In order to simplify the model, it is assumed that the auction interval is very short and is ignored. The complete reverse auction-based offloading algorithm is illustrated in Algorithm 2.

**3.2.1. Allocation.** In the allocation steps, the mobile device decides which channel will be the auction winner. In order to avoid the extra delay caused by the multiround auction, the single-round auction is implemented in this paper. Thus, jointly considering the resources and the price that the bidders can provide, the mobile device decides who will win the auction, and  $b_j$  is the transaction price. Therefore, given the abovementioned definitions and notation, the optimization problem can be converted into the reverse auction problem. Here,  $\psi_{i,j}$  represents the consequence of auction.  $\psi_{i,j} = 0$  denotes that there is no winner channel. On the contrary,  $\psi_{i,j} = 1$  expresses that the  $j$ th channel wins the auction. Our goal is to maximize the utility of the mobile device user, which can be formulated as

$$\max_{\{\psi_{i,j}\}} \mathcal{F} = \sum_{i=1}^{\mathcal{N}} \mathcal{P}_i - \sum_{j=1}^{\mathcal{M}} \sum_{i=1}^{\mathcal{N}} \left( (1 - \psi_{i,j}) \mathcal{P}_i + \psi_{i,j} b_j \right). \quad (19)$$

In order to determine the winner and the allocation relationship, the bid densities of the participants are calculated and sorted firstly. In the list of wireless channels, the wireless channels were ranked in ascending order of their bid densities. For mobile users, the lowest call density is the

best communication quality. The bid density of sellers can be calculated by

$$bd_j = \frac{(\mathcal{E}_j - \sum_{r=1, r \neq i}^{\mathcal{N}} \psi_{r,j} P_i^{\text{tr}} G_i) E_{i,j}^{\text{Off}}}{\sqrt{\mathcal{E}_j - \sum_{r=1, r \neq i}^{\mathcal{N}} \psi_{r,j} P_i^{\text{tr}} G_i}}, \quad (20)$$

where  $\mathcal{E}_j - \sum_{r=1, r \neq i}^{\mathcal{N}} \psi_{r,j} P_i^{\text{tr}} G_i > 0$ , which is an indispensable condition for the wireless channel to ensure their quality of service. If the value is less than or equal to zero, the channel will give up participating in the auction.

**3.2.2. Pricing Model.** The final transaction price paid by the mobile device is  $b_j$ , which is the bid price submitted by the winner wireless channel. The utility of the mobile device user can be formulated as

$$\mathcal{F} = \sum_{i=1}^{\mathcal{N}} \mathcal{P}_i - \sum_{j=1}^{\mathcal{M}} \sum_{i=1}^{\mathcal{N}} \left( (1 - \psi_{i,j}) \mathcal{P}_i + \psi_{i,j} b_j \right). \quad (21)$$

If the mobile user does not participate in the auction, its utility value is equal to 0. In other words, if  $\psi_{i,j} = 0$ , obviously then  $\mathcal{F} = 0$  through the calculation of formula (21). Moreover, the utility of the wireless channel can be formulated as

$$\Theta = \sum_{j=1}^{\mathcal{M}} \sum_{i=1}^{\mathcal{N}} \psi_{i,j} b_j. \quad (22)$$

If the wireless channel does not win the auction, then  $\psi_{i,j} = 0$ , obviously the utility of the wireless channel is equal to zero.

3.2.3. *Properties.* In this section, the properties of the proposed reverse auction model are analyzed. The individual rationality and the truthfulness properties need to be proved.

- (1) Individual rationality: when the utility of each participating bidder in the pricing stage is greater than zero, the proposed mechanism is individual rational for each winning bidder. Namely,

$$\mathcal{F}_{B_M \setminus \{b_j\}} - (\mathcal{F}_{B_M} - \mathcal{F}_{b_j}) \geq 0, \quad (23)$$

where  $B_M = \{b_1, b_2, \dots, b_j, \dots, b_M\}$  and  $\mathcal{F}_{B_M \setminus \{b_j\}}$  denotes the utility of the mobile device under the optimal allocation solution without the presence of the  $j$ th channel.

- (2) Truthfulness: for each bidder, the truthfulness means that the bid price of each bidder is equal to its private value. If the bidding of channels is untrue, the utility will be unlikely the biggest. In order to get the maximum, the allocation should be formulated as follows:

$$\mathcal{F}_{B_M \setminus \{b_j\}} - (\mathcal{F}_{B_M} - \mathcal{F}_{b_j}) \geq 0, \quad (24)$$

$$\begin{aligned} \Omega &= \mathcal{F}_{B_M \setminus \{b_j\}} - (\mathcal{F}_{B_M} - \mathcal{F}_{b_j}) - \left[ \mathcal{F}_{B_M \setminus \{b_i\}} - (\mathcal{F}_{B_M} - \mathcal{F}_{b_i}) \right] \\ &= (\mathcal{F}_{B_M \setminus \{b_j\}} + \mathcal{F}_{b_j}) - (\mathcal{F}_{B_M \setminus \{b_i\}} + \mathcal{F}_{b_i}) \end{aligned} \quad (25)$$

Based on the proposed reverse auction mechanism in this paper, because the bid price of the channel is not greater than the reverse price of the mobile device user,  $\Omega \leq 0$ . Obviously, when  $j = i$ , the value of  $\Omega$  is equal to zero. Therefore, each bidder must be truthful to obtain the maximum utility.

## 4. Simulation and Analysis

In this section, the performance of the proposed mechanism is evaluated through numerical simulations designed by using the MATLAB. The compared algorithms are the competition-based algorithm [30] and the user-satisfaction-based offloading algorithm [31]. Their features are described as follows:

- (1) Competition-based algorithm: the system is modeled as a competitive game subjected to the job execution deadlines and user-specific channel bit rates. Each user tries to minimize its own energy consumption when it competes for the shared communication channel. The Gauss–Seidel-like method is executed for achieving the Nash equilibrium to derive the mobile device user's offloading decisions.
- (2) User-satisfaction algorithm: a utility function is introduced to choose the best communication resources in terms of user-satisfaction parameters, such as the throughput, used energy, and time spent to execute the application. Based on this, the offloading strategy is obtained by the applications' computation percentage.

Without loss of generality, four performance metrics of the proposed algorithm and the two classical algorithms are

compared on the same simulation scenarios fairly. The four metrics are the average energy consumption, delay, energy efficiency factor, and throughput of the mobile device for offloading.

4.1. *Simulation Setup.* The simulations are deployed based on real-world settings. All the parameters, including the energy consumption rates and computing capacity, are measured from real mobile devices. These real-world datasets, which have been widely used, are measured at various clock speeds and in the cellular network scenarios by using a monsoon power monitor.

At first, a base station is considered that covers a hexagonal cellular network with radius 2 km, and assume that the wireless access point is located at the center. The base station has  $M = 4$  channels, and the channels belonging to this base station are orthogonal. The bandwidth capacity of the channels can be different values, but in order to simplify the simulation, four channels of the same bandwidth of the device are set to  $w = 1$  MHz, which does not affect the effect of the experiment. Besides, the power of the background noise is set to  $\sigma^2 = -100$  dBm, and the path loss factor is set to  $a = 2$  according to the physical interference model. In the system of mobile-edge cloud computing, mobile devices are randomly distributed in the coverage area of the hexagonal cellular network, accessing to this wireless point at any time if needs. And there is a mobile-edge server deployed near the base station, who assigns 5 GHz computation capability for each mobile device, sufficient to satisfy the requirements of all mobile devices.

Conforming to the diversity of the mobile device in the real world, four types of smartphones are considered, namely, Galaxy Note, Galaxy Note 2, Nexus S, and HP iPAQ PDA. Different mobile devices have different CPU computing capacities. The HP iPAQ PDA with a 400 MHz Intel XScale processor [31] has the following parameters: the local processing power  $P_i^l = 0.9$  W, the standby power  $P_i^{id} = 0.3$  W, and the transmission power  $P_i^{tr} = 1.3$  W. In addition, the parameters of the other three mobile devices include CPU processing parameters, such as  $\chi_i$ ,  $\alpha_i$ , and  $\beta_i$ . These parameters are adopted as in [30]. In the simulation, the type of the mobile device in the mobile-edge cloud computing scenario is randomly selected among the abovementioned three types, and each mobile device has only one task waiting to be executed. The tasks on mobile devices are set to ten types: face recognition, virus scanning, online gaming, and so on. These ten types of tasks are randomly assigned to each mobile user. Different types of mobile devices have different processing speeds for different task types, whose corresponding parameters are given in Table 2, including workload density, data size, and the allocated computing capacity.

It is clear that, in these tasks, the workload densities of face recognition and virus scanning are larger than those of other types of tasks, and the data size of the two tasks is relatively small, which are computation-intensive tasks. On the contrary, the workload density of video coding is far less than that of the other eight tasks, but the data size is

TABLE 2: Parameters of the system.

Smartphone	$\chi_i$	$\alpha_i$	$\beta_i$	$P_i^r$	$P_i^d$
Galaxy Note	3.0	0.33	0.10	2605	9.64
Galaxy Note 2	2.7	0.25	0.40	2796	11.70
Nexus S	3.0	0.34	0.35	1217	7.4
Galaxy Nexus	3.0	0.40	0.30	964	22.37
Task	$O_i$	$D_i$	$C_i^l$	—	—
Face recognition	60	31680	1.2	—	—
400-frame game	2048	2640	1.0	—	—
Chess: select	400	1580	0.6	—	—
Chess: move	400	2640	1.0	—	—
Virus scanning	300	32946	1.5	—	—
4-queen puzzle	200	87.8	0.4	—	—
5-queen puzzle	200	263	0.45	—	—
6-queen puzzle	200	1760	0.72	—	—
7-queen puzzle	200	8250	1.04	—	—
Video transcoding	10240	200	0.56	—	—

particularly larger than that of others, which belong to communication-intensive tasks. It is obvious that the parameters given in the table include various types of tasks that satisfy the generality and credibility of the simulation.

In order to accurately evaluate the performance of the algorithm without any loss of generality, a series of simulations are carried out, gradually increasing the number of mobile devices from 50 to 1000. Since the mobile devices are randomly deployed within the coverage of mobile networks, and the type of the mobile device and the task request of mobile users have stochastic features.

**4.2. Evaluation Results.** Firstly, the energy consumption of the proposed algorithm is evaluated. Figure 3 shows the average energy consumption of the mobile device when the number of mobile devices increases from 50 to 1000 with four different methods. The average energy consumed by one mobile device is approximately 21.2060 J with the local computing approach. Comparing with the local computing approach, both the proposed approach and the other two algorithms achieve the purpose of energy saving through task offloading.

At the beginning, with 50 mobile devices, the three methods exhibit an energy consumption of 6.4219 J, 6.4276 J, and 6.9430 J, respectively. With the gradually increased number of mobile devices, the average energy consumption of the mobile device increases to 11.0077 J, 12.5876 J, and 13.3540 J, respectively. This is because too many mobile devices choose to access the same wireless channel to implement the task offloading simultaneously, which would lead to the augment of mutual interference. According to (4), it is obvious that the severe interference to each other will cause the reduction of the communication quality and the rates for computation offloading. Therefore, with 1000 mobile devices, more and more users tend to choose the local computing method, and the average energy consumption of

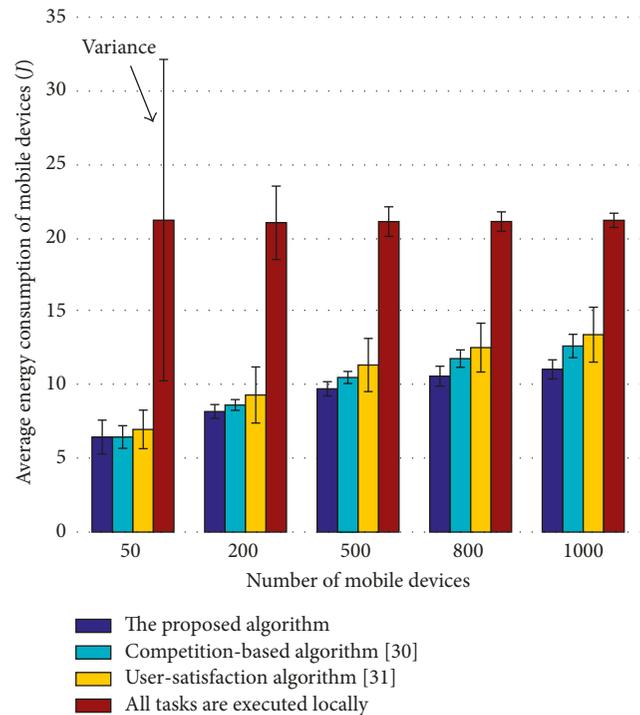


FIGURE 3: Average energy consumption of mobile devices.

mobile devices increases. The proposed mechanism can save at least 56.42% of the energy consumption.

The superiority of the proposed approach is gradually obvious. This is due to the fact that the reverse auction-based offloading mechanism performs task offloading decision in a global long-term perspective, reasonably allocating communication resources for mobile device users to meet the quality of service requirements. It exhibits a relatively lower energy consumption when the number of mobile device users is small. However, with the explosive increase in the

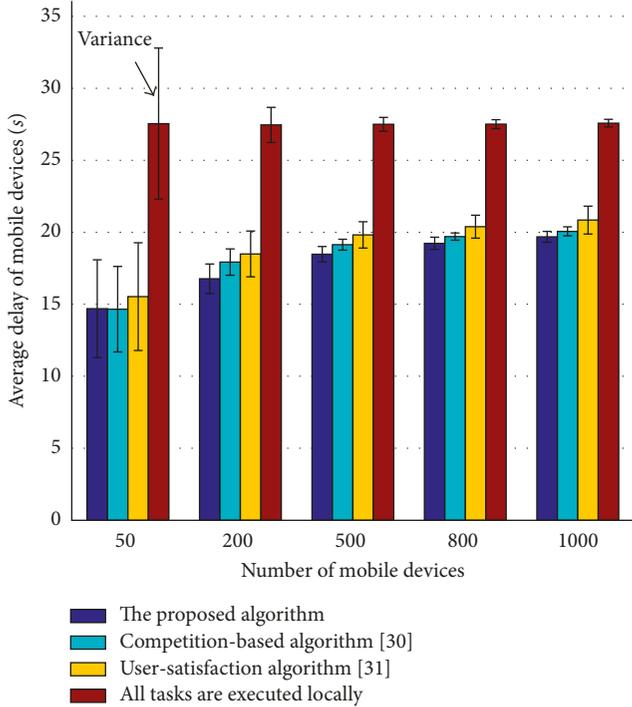


FIGURE 4: Average delay of mobile devices.

number of mobile device users, the performance degrades due to the traffic growth. Obviously, the proposed method can find a better energy-saving solution than other two approaches.

The average task execution delay of mobile devices with the proposed method and the other two schemes is compared. As shown in Figure 4, the average time delay of mobile devices for performing a task is approximately 27.5157 s with the local computing approach. With 50 mobile devices, the time delays of these three methods are 14.6868 s, 14.6605 s, and 15.5306 s, respectively. Compared with the local computing approach, at least 45.69% of the time delay can be saved. When 1000 mobile devices are deployed, the time delays obtained by four methods are 19.6782 s, 20.0610 s, 20.8447 s, and 27.5157 s, respectively. The proposed mechanism can save about 35.43% of the time compared to the local computing approach, which is slightly higher than the performance of other two algorithms.

Third, the throughput of mobile devices is compared in the case of our proposed method and other two methods in addition to the local computing method because the local processing does not need to upload data and the throughput is zero. Figure 5 shows that, at the beginning, with 50 mobile device users, the other three methods exhibit an average throughput of 5144.6 bps, 5883.7 bps, and 6195.7 bps, respectively. Although the throughput of mobile devices in the case of our proposed algorithm is lower at the beginning, when the number of mobile devices is between 50 and 200, the trend of throughput drops more slowly than the other two methods. With the continued growth of mobile devices, the throughput of mobile devices in the case of the proposed method is higher than that in the other methods. At the end, with 1000 mobile device users, the methods exhibit an

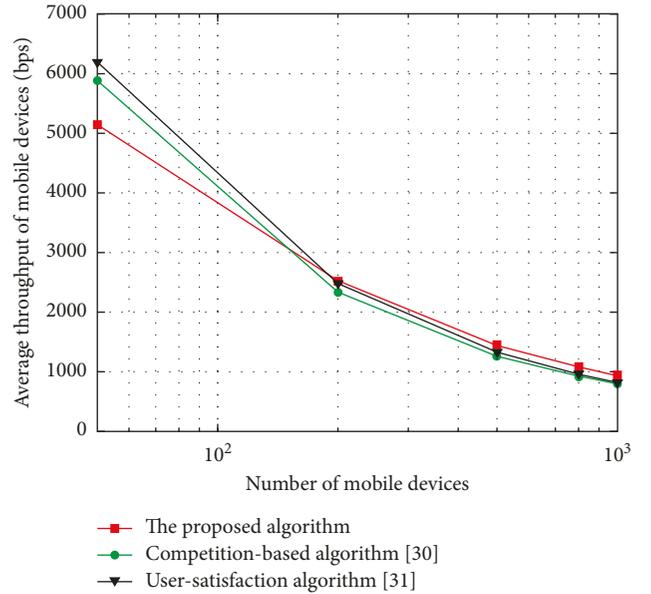


FIGURE 5: Average throughput of mobile devices.

average throughput of 932.3 bps, 796.4 bps, and 819.5 bps, respectively. As the number of mobile devices increases logarithmically, the correspondingly mutual interference among the device will grow. Furthermore, the uplink data transmission rate will decrease, which leads the energy consumption of cloud offloading greater than that of local computing. Thus, more and more mobile device users will adopt local computing, substituting for offloading operation. Compared with the competition-based algorithm and the user-satisfaction algorithm, the throughput is higher and the rate of decline is relatively slow when using the proposed method.

Finally, the energy efficiency factor for offloading is evaluated with the proposed method, competition-based algorithm, and user-satisfaction algorithm over 1000 simulation runs. The proposed mechanism is designed to reduce the energy consumption and the response time delay of mobile devices. Thus, a function is proposed representing the QoS degree perceived by the user. The function is modeled as a sigmoid curve, which is widely used to measure user satisfaction and service quality in previous studies [32]. User satisfaction increases as energy consumption and latency decrease, so we use sigmoid functions to represent the relationship between them:  $f_1 = 1 - (1/1 + e^{-(E^{\text{aver}} - E^t)})$ ;  $f_2 = 1 - (1/1 + e^{-(T^{\text{aver}} - t^t)})$ .

The function  $\mathcal{U} = \omega_1 f_1 + \omega_2 f_2$  is introduced to analyze the energy efficiency factor, where  $\omega_1 + \omega_2 = 1$ . Moreover,  $\omega_1$  and  $\omega_2$  represent the weight coefficients of energy consumption and delay, respectively. And  $E^{\text{aver}}$  and  $T^{\text{aver}}$ , respectively, denote the average energy consumption and average delay. As shown in Figure 6, with an increased number of mobile devices, the user satisfaction for task offloading gradually reduced. And at the last, with 1000 mobile devices, the values of the two methods of comparison drop sharply. On the contrary, compared with the other two curves, the curve corresponding to the proposed method is relatively stable. Therefore, when there are a large number of

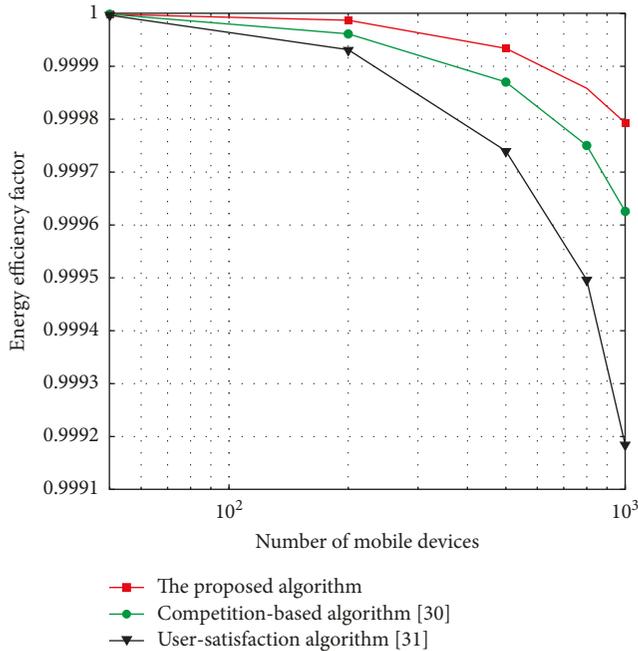


FIGURE 6: Energy efficiency factor of mobile devices.

mobile devices in the mobile-edge cloud system, the performance of the proposed method is better to meet the request of mobile device users for cloud offloading.

## 5. Conclusion

In this paper, an energy-aware task offloading mechanism is designed to perform offloading decisions, with optimization on minimizing the energy consumption of mobile devices. Considering the interference threshold in each channel, the task local execution delay, and the local energy consumption, the task offloading decision problem is formulated as a 0-1 nonlinear integer programming optimization. In order to solve this problem, the algorithm is proposed for classifying the mobile device and priority determination. Furthermore, the reverse auction theory has been implemented with the proposed algorithm to decide the offloading target channel. The individual rationality and truthfulness of the reversed auction model are also discussed in the paper. The performances of the proposed mechanism, comparing with the other two methods, are evaluated with performance metrics of energy consumption, time delay, throughput, and the energy efficiency factor. The simulation results validate that the proposed algorithm can achieve better performances.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant nos. 61379111, 61402538, 61403424, 61502055, 61672537, and 61672539).

## References

- [1] R. Janessa and R. Meulen, *Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units by 2020*, Gartner Inc., Stamford, CT, USA, 2013.
- [2] CISCO, *The Internet of Things: How the Next Evolution of the Internet is Changing Everything*, CISCO White Paper, 2011.
- [3] E. Ahmed, A. Gani, M. K. Khan, R. Buyyac, and S. U. Khan, "Seamless application execution in mobile cloud computing: motivation, taxonomy, and open challenges," *Journal of Network and Computer Applications*, vol. 52, pp. 154–172, 2015.
- [4] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.
- [5] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 45–55, 2014.
- [6] S. Abolfazli, Z. Sanaei, E. Ahmed et al., "Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 337–368, 2014.
- [7] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2014.
- [8] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *Proceedings of the IEEE 14th Workshop on Signal Processing Advances in Wireless Communications*, pp. 26–30, Darmstadt, Germany, June 2013.
- [9] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transaction on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.
- [10] L. Yang, J. Cao, H. Cheng, and J. Yusheng, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2253–2266, 2015.
- [11] H. Viswanathan, E. K. Lee, I. Rodero, and D. Pompili, "Uncertainty-aware autonomic resource provisioning for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2363–2372, 2015.
- [12] O. Munoz-Medina, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4738–4755, 2015.
- [13] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.
- [14] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, 2015.
- [15] W. Li, Y. Zhao, S. Lu, and D. Chen, "Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 89–97, 2015.
- [16] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 34–44, 2013.

- [17] 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. 99, 2017.
- [18] S. Wang, R. Urgaonkar, M. Zafer, and T. He, "Dynamic service migration in mobile edge-clouds," in *Proceedings of the IFIP Networking Conference*, pp. 1–9, Toulouse, France, March 2015.
- [19] 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. 4, pp. 974–983, 2015.
- [20] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal and Information Processing Over Networks*, vol. 1, no. 2, pp. 89–103, 2015.
- [21] M. T. Beck and M. Maier, "Mobile Edge Computing: Challenges for Future Virtual Network Embedding Algorithms," *The Eighth International Conference on Advanced Engineering Computing and Applications in Sciences*, pp. 65–70, Rome, Italy, 2014.
- [22] Y. Zhang, C. Lee, D. Niyato, and P. Wang, "Auction approaches for resource allocation in wireless systems: a survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1020–1041, 2013.
- [23] B. Kollimarla, *Spectrum Sharing in Cognitive Radio*, College of Oklahoma State University, Oklahoma City, OK, USA, 2009.
- [24] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "A double-auction mechanism for mobile data-offloading markets," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1634–1647, 2015.
- [25] S. Paris, F. Martignon, I. Filippini, and L. Chen, "An efficient auction-based mechanism for mobile data offloading," *IEEE Transactions on Mobile Computing*, vol. 14, no. 8, pp. 1573–1586, 2015.
- [26] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510–2523, 2015.
- [27] K. Son and B. Krishnamachari, "SpeedBalance: speed-scaling-aware optimal load balancing for green cellular networks," in *Proceedings of the IEEE INFOCOM 2012*, pp. 2816–2820, Orlando, FL, USA, March 2012.
- [28] M. Xiao, N. B. Shroff, and E. K. P. Chong, "A utility-based power-control scheme in wireless cellular systems," *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 210–221, 2003.
- [29] M. Chiang, P. Hande, T. Lan, and C. W. Tan, "Power control in wireless cellular networks," *Foundations and Trends in Networking*, vol. 2, no. 4, pp. 381–533, 2008.
- [30] E. Meskar, T. Todd, D. Zhao, and G. KarakLondon UKostas, "Energy efficient offloading for competing users on a shared communication channel," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 3192–3197, London UK, June 2015.
- [31] D. Mazza, D. Tarchi, and G. E. Corazza, "A user-satisfaction based offloading technique for smart city applications," in *Proceedings of the 2014 IEEE Global Communications Conference*, pp. 2783–2788, Austin, TX, USA, December 2014.
- [32] D. H. V. Seggern, *CRC Standard Curves and Surfaces with Mathematica*, CRC Press, Boca Raton, FL, USA, 2015.



Hindawi

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

