Hindawi
Mobile Information Systems
Volume 2019, Article ID 3613250, 9 pages
https://doi.org/10.1155/2019/3613250

Research Article

Energy-Efficient Computation Offloading and Transmit Power Allocation Scheme for Mobile Edge Computing

Xiaohui Gu, Li Jin, Nan Zhao, and Guoan Zhang

School of Information Science and Technology, Nantong University, Nantong 226019, China

Correspondence should be addressed to Guoan Zhang; gzhang@ntu.edu.cn

Received 30 April 2019; Accepted 12 August 2019; Published 16 December 2019

Academic Editor: Marco Anisetti

Mobile edge computing (MEC) is considered a promising technique that prolongs battery life and enhances the computation capacity of mobile devices (MDs) by offloading computation-intensive tasks to the resource-rich cloud located at the edges of mobile networks. In this study, the problem of energy-efficient computation offloading with guaranteed performance in multiuser MEC systems was investigated. Given that MDs typically seek lower energy consumption and improve the performance of computing tasks, we provide an energy-efficient computation offloading and transmit power allocation scheme that reduces energy consumption and completion time. We formulate the energy efficiency cost minimization problem, which satisfies the completion time deadline constraint of MDs in an MEC system. In addition, the corresponding Karush–Kuhn–Tucker conditions are applied to solve the optimization problem, and a new algorithm comprising the computation offloading policy and transmission power allocation is presented. Numerical results demonstrate that our proposed scheme, with the optimal computation offloading policy and adapted transmission power for MDs, outperforms local computing and full offloading methods in terms of energy consumption and completion delay. Consequently, our proposed system could help overcome the restrictions on computation resources and battery life of mobile devices to meet the requirements of new applications.

1. Introduction

For the past several years, as mobile devices (MDs) such as smartphones, handheld game consoles, and vehicle multimedia computers have become virtually ubiquitous, an increasing number of new mobile applications such as augmented reality, image processing, natural language processing, face recognition, and interactive gaming have emerged and become the focus of considerable attention [1, 2]. These types of mobile applications are typically latency-sensitive, demand intensive computation, and have high-energy consumption characteristics. With the constraint of physical size, MDs usually have limited resources, which restricts their battery life and computation capacities [3, 4].

Recent studies have shown that mobile edge computing offloading (MECO) technology provides a promising opportunity to effectively overcome the limitations related to the hardware and energy consumption problems of MDs by offloading computation-intensive tasks to adjacent clouds at the edges of mobile networks for execution [5–7]. In particular, mobile edge computing (MEC) offers cloud computing capabilities at the very edge of mobile networks by implementing MEC servers at base stations (BSs), with short latency and excellent quality of experience for mobile users, and it has drawn significant attention from both academia and industry [8].

Given that computation performance and energy consumption are of great importance to mobile users [9], it is essential to design effective computation offloading schemes for MECO systems. To minimize the completion time of the computing tasks for MDs, Hong et al. [10] formulated a joint optimization problem for the time division multiple access (TDMA) and frequency-division multiple access (FDMA) schemes in a multiuser MECO system. Liu et al. [11] derived a power-constrained delay minimization offloading policy in MEC systems, in which computation task scheduling is considered, and the Markov decision process method was adopted to solve the proposed optimization problem. An effective computation offloading scheme was also presented by Mao et al. [12], with the aim of reducing the execution time in a green MEC system with energy harvesting devices. In a multiuser TDMA (MU-TDMA) MECO system, Ren et al. [6] studied the latency minimization problem through the joint allocation of communication and computation resources, and the minimum system delay for both the local and edge computing models was derived. However, the deficiency of the abovementioned works is that the energy consumption at the MD side was not taken into account by the offloading decision.

Rapid battery depletion has also posed a significant obstacle in contemporary networks. In response, Sardellitti et al. [13] designed an energy-minimization offloading problem that optimizes the radio resources in an MIMO multicell system. Zhang et al. [14] presented an energy-efficient computation offloading scheme that incurs minimal energy consumption under latency constraints by optimizing the offloading policy and radio resource allocation for MEC in 5G heterogeneous networks with multiaccess characteristics. You et al. [15], based on insight into the input data arrival time instants and computation deadlines, studied an energy-efficient resource management policy for MECO systems and formulated an optimization strategy that minimizes the total mobile-energy consumption. Unfortunately, these works are mainly interested in reducing the energy consumption without attempting to reduce the completion time of computation tasks.

Moreover, MDs are utilized by various individuals, and they may pursue diverse interests. Therefore, it is necessary to focus on both energy and time consumption when designing the offloading policy. In recent years, a few works have considered the tradeoff between the energy consumption at MDs and the execution delay in MECO systems [16, 17]. However, in these optimization models, only the transmission delay was considered and the server calculation delay was omitted; thus, they cannot be applied to an MEC server with limited computing capacity.

Most of the above works for multiuser MECO systems concentrated only on the binary computation offloading strategy, which implies that the computation task is executed via either local computing or edge computing. However, according to their communication capacity, some MDs may prefer partial offloading. By offloading time-consuming and/or energy-consuming subtasks to the MEC servers, this partial offloading can achieve higher energy savings and lower computation latency compared with those of binary offloading [18]. Although the initiative work in [19] studied the energy saving partial computation offloading problem, a more pressing design objective for some latency-intensive applications has not been discussed, namely, the latency minimization issue. In addition, Guo et al. [20] proposed an energy-efficient dynamic binary offloading and resource scheduling (eDors) policy and designed a distributed eDors algorithm to minimize the energy efficiency cost (EEC), which is defined as the weighted sum of energy consumption and computation completion time of a task. With the above observations, in this study, an EEC minimization problem was investigated for a multiuser MEC system with partial computation offloading. Moreover, a special case of binary offloading, which refers to full offloading or complete local computing, is also discussed herein. This study was conducted to minimize the EEC paid by the MD for completing a task with respect to the constraint of completion time deadline. Specifically, it is also proved that the EEC minimization problem is convex, and we are able to solve the convex optimization problem by employing the Karush–Kuhn–Tucker (KKT) conditions. Furthermore, an optimal computation offloading and transmit power allocation scheme is presented according to the EEC on the MEC server and the local device. The main contributions of this paper are summarized as follows:

(i) We present a multiuser computation offloading framework for MEC and address the problem of performance-guaranteed computation offloading.

(ii) An EEC optimization problem for minimizing the weighted sum of the energy consumption and computation completion time while satisfying the latency constraint is formulated.

(iii) The Lagrange multiplier method and the KKT conditions are utilized to solve the convex optimization problem, and an efficient algorithm consisting of computation offloading policy and transmission power allocation for the MDs is presented.

The remainder of this paper is organized as follows. The ensuing section presents the system model. Next, the optimization problem is formulated. Subsequently, the proposed effective task offloading algorithm is described. Then, numerical results are presented to demonstrate the excellent performance of our proposed method compared to that of the existing methods. Finally, concluding remarks are presented.

2. System Model

The considered MEC system consists of N MDs, as shown in Figure 1. The MEC server is a computing device installed at the wireless access station. The MDs can connect to the station resource, which is located in proximity to the mobile users. Assigning computing tasks to the base stations (BS) can help mobile users improve computing performance. As pioneering literatures already examined mobile cloud computing (e.g., [10, 15, 21]) and mobile networking (e.g., [22, 23]), and to enable tractable performance analysis and obtain useful insights, we consider the application scenario as quasistatic, where the set of MDs will not be changed during a computation offloading period. A set of $\mathcal{N} = \{1, 2, \ldots, N\}$ collocated MDs is considered in this study, and a computation-intensive task is set for each MD to be completed. Let the tuple $\{c_n, d_n\}$ denote the task requirement of MD n, where c_n describes the CPU circles required to complete the task and d_n presents the task data size. The offloading data size of MD n is denoted as l_n. Let α_n represent the fraction of the offloading task for MD n with

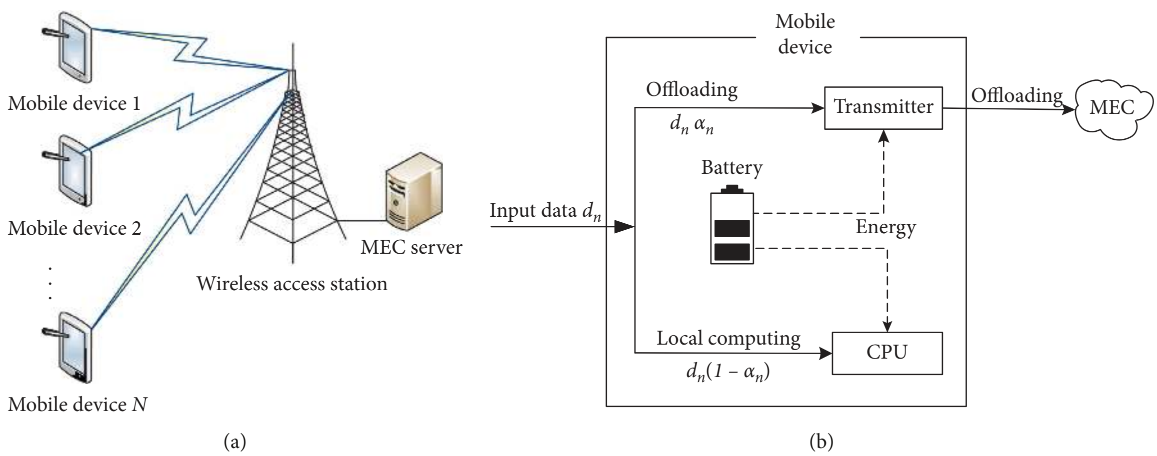

Figure 1: System model. (a) Multiuser mobile edge computing system. (b) Architecture of an MD.

the condition $l_n = d_n\alpha_n$. The offloading policy profile of all MDs is denoted by $\mathscr{A} = \{\alpha_n \mid n \in N\}$.

2.1. Communication Model. We first introduce the communication model for the MEC system. The MD makes the computation offloading policy based on its energy consumption and completion time performance. The transmission power for MD n is denoted as p_n, and g_n represents the channel gain of the BS. Furthermore, we consider a multiuser computing offloading system in this paper, they will interfere each other in the uplink. Thus, the uplink data rate for computation offloading of MD n is given by

$$r_n = B \log_2\left(1 + \frac{p_n g_n^2}{(N_0 + I)B}\right), \quad (1)$$

where N_0 and I represent the power spectrum density of additive White Gaussian noise and interference, respectively. Letting B to be the bandwidth of the channel, at the MEC, the received signal power can be denoted by a function of data rate r:

$$h(r) = (N_0 + I)B\left(2^{(r/B)} - 1\right), \quad (2)$$

which is monotonically increasing and convex for $r > 0$. The offloading transmission rate can be denoted as

$$r_n = \frac{d_n}{t_n}, \quad (3)$$

where t_n is the transmission time of MD n for offloading of the input data of size d_n. Then, the transmission power p_n can be calculated by combining (2) and (3):

$$p_n = \frac{1}{g_n^2} h\left(\frac{d_n}{t_n}\right). \quad (4)$$

2.2. Computation Model. Consider that MD n has a computation task $\mathscr{T}_n = \{c_n, d_n\}$, where c_n denotes the total number of CPU cycles required to accomplish the computation task $\mathscr{T}_n$ and d_n describes the input data size of computation task $\mathscr{T}_n$. Next, we will discuss the EEC spent by the MDs with respect to energy consumption and completion time for the local computing and edge computing approaches, respectively.

2.2.1. Local Computing. With the local computing approach, MD n executes its computation task $\mathscr{T}_n$ locally on the MD. Let h_n denote the computation capability (i.e., CPU cycles per second) of MD n; different MDs may have different computation capabilities. Accordingly, the completion time for local computing is defined as

$$t_{n,\mathrm{loc}} = \frac{c_n}{h_n}. \quad (5)$$

For the computational energy, we have

$$e_{n,\mathrm{loc}} = f_n c_n, \quad (6)$$

where f_n is the consumed energy per CPU cycle for MD n.

According to equations (5) and (6), the EEC for the local computing approach in terms of computational time and energy is computed by

$$Z_{n,\mathrm{loc}} = \gamma_n^e e_{n,\mathrm{loc}} + \gamma_n^t t_{n,\mathrm{loc}}, \quad (7)$$

where $\gamma_n^e, \gamma_n^t \in [0, 1]$ denote the weights of energy consumption and computation completion time for MD n to make the offloading policy on task $\mathscr{T}_n$, respectively. We permit the MDs to set different weight values during policy making to meet their specific requirements. For example, a device with a lower battery energy is more likely to choose a larger γ_n^e when making the offloading policy to save more energy. When an MD is running delay-sensitive applications (e.g., online games), it aims to set a larger γ_n^t to reduce the latency.

2.2.2. Edge Computing. With the edge computing approach, MD n offloads its computation task $\mathscr{T}_n$ to the MEC server. Subsequently, the server performs the computation task and feeds the results back to the mobile. Clearly, MD n offloads its computing task $\mathscr{T}_n$ for execution in the MEC server, including three consecutive phases: (i) transmitting phase, (ii) computing phase, and (iii) receiving phase.

According to the communication model, the transmission time and energy consumption of MD n for transmitting its computation task $\mathcal{T}_n$ to the MEC server are, respectively, calculated by

$$t_{n,\mathrm{trs}} = \frac{d_n}{r_n} = t_n, \tag{8}$$

$$e_{n,\mathrm{trs}} = p_n t_n. \tag{9}$$

Furthermore, the computation execution time of task $\mathcal{T}_n$ on the MEC server is computed by

$$t_{n,\mathrm{exe}} = \frac{c_n}{h_n^c}, \tag{10}$$

where h_n^c denotes the computation capacity of the MEC server. In this study, we considered the energy consumption on the MDs side, whereas our future work will consider the execution energy consumption of the MEC server. Therefore, for the edge computing approach, the completion time and energy consumption of task $\mathcal{T}_n$ are, respectively, denoted as

$$t_{n,\mathrm{off}} = t_{n,\mathrm{trs}} + t_{n,\mathrm{exe}} + t_{n,\mathrm{rece}}, \tag{11}$$

$$e_{n,\mathrm{off}} = e_{n,\mathrm{trs}} + e_{n,\mathrm{rece}}, \tag{12}$$

where $t_{n,\mathrm{rece}}$ and $e_{n,\mathrm{rece}}$ denote the time and energy required by MD n when receiving the computation outcome from the MEC server, respectively. From equations (9) and (11), the EEC for the edge computing approach can be calculated by equation (13) based on completion time and energy consumption.

$$Z_{n,\mathrm{off}} = \gamma_n^e e_{n,\mathrm{off}} + \gamma_n^t t_{n,\mathrm{off}}. \tag{13}$$

It is observed from equations (8) and (9) that a low data transmission rate r_n of MD n would result in a long transmission time and high energy consumption during wireless access when offloading input data to the MEC server. As in the existing works [17, 24], the receiving time $t_{n,\mathrm{rece}}$ and the receiving energy $e_{n,\mathrm{rece}}$ can be ignored, because for many applications, such as face recognition, the size of the result is typically much smaller than that of the input data.

3. Problem Formulation

In this section, an EEC optimization problem for the MEC system is formulated by jointly considering the energy consumption and task completion time for each MD. Here, α_n is defined as the part of task $\mathcal{T}_n$ offloaded to the MEC server. Therefore, the EEC of MD n includes both local computing consumption and offloading consumption:

$$Z_n = Z_{n,\mathrm{off}}\alpha_n + Z_{n,\mathrm{loc}}(1-\alpha_n). \tag{14}$$

The completion time of task $\mathcal{T}_n$ of MD n can be denoted as

$$t_{n,\mathrm{all}} = t_{n,\mathrm{off}}\alpha_n + t_{n,\mathrm{loc}}(1-\alpha_n). \tag{15}$$

The objective is to provide the optimal computation offloading policy $\mathcal{A}^*$ and transmission power allocation $\mathcal{P}^*$ to minimize the EEC. Hence, the EEC for all MDs can be formulated as a constrained minimization problem:

$$\begin{aligned} \min_{\mathcal{A},\mathcal{P}} \quad & \sum_{n=1}^{N} Z_n, \\ \text{s.t.} \quad & \left(t_n + \frac{c_n}{h_n^c}\right)\alpha_n + \frac{c_n}{h_n}(1-\alpha_n) \le T_{n,\max}, \quad \forall n, \end{aligned} \tag{16}$$

where $\mathcal{A} = \{\alpha_n \mid n \in N\}$, $\mathcal{P} = \{p_n \mid n \in N\}$. As expressed in equation (4), p_n^* can be conducted from t_n^*. Therefore, the variables of the optimization function (16) are α_n and t_n. The constraint specifies that the total completion time of the task of MD n is bounded by the required maximum completion time $T_{n,\max}$. The convexity of the optimization problem in equation (16) is explored as follows.

Proof. First, it should be proved that the objective function Z_n in equation (16) is jointly convex with respect to the optimization variables α_n and t_n. Then, we show the convexity of the constraint.

$$\begin{aligned} Z_n &= \left(\gamma_n^e e_{n,\mathrm{off}} + \gamma_n^t t_{n,\mathrm{off}}\right)\alpha_n + \left(\gamma_n^e e_{n,\mathrm{loc}} + \gamma_n^t t_{n,\mathrm{loc}}\right)(1-\alpha_n) \\ &= \left[\gamma_n^e \frac{t_n}{g_n^2} h\left(\frac{d_n}{t_n}\right) + \gamma_n^t\left(t_n + \frac{c_n}{h_n^c}\right)\right]\alpha_n \\ &\quad + \left(\gamma_n^e f_n c_n + \gamma_n^t \frac{c_n}{h_n}\right)(1-\alpha_n), \\ \frac{\partial^2 Z_n}{\partial t_n^2} &= \frac{2^{(d_n/Bt_n)} d_n^2 \alpha_n r_n^e (\ln 2)^2 N_0}{B g_n^2 t_n^3} \ge 0. \end{aligned} \tag{17}$$

As Z_n is an affine function of α_n, it is convex with respect to the optimization variable α_n. Similarly, the constraint is jointly convex with respect to the optimization variables α_n and t_n. It can be seen that the optimization problem in equation (16) has a zero-duality gap and satisfies the Slater constraint condition. The result of the zero-duality gap provides a manner to obtain the optimal solution of the original problem in equation (16) derived from its corresponding duality problem.

4. Algorithm for Minimum Energy Efficiency Cost (EEC) Problem

By solving the dual problem of equation (16), the computation offloading and resource allocation scheme is derived. Therefore, the Lagrange function of the primal problem in equation (16) is defined as $L(\alpha_n, t_n, \lambda)$. The Lagrange multiplier λ denotes the price at which the total completion time of the task of MD n does not exceed the required maximum completion time.

The dual problem for the primal problem in equation (16) is given by

$$\max_{\lambda}\min_{\alpha_n,t_n} L(\alpha_n, t_n, \lambda), \tag{18}$$

where $\lambda \geq 0$ is the Lagrange multiplier, which is associated with the completion time constraint. Then, the corresponding KKT condition is applied to transform it into the following equations:

$$\begin{aligned} L(\alpha_n, t_n, \lambda) &= \left[\gamma_n^e \frac{t_n}{g_n^2} h\left(\frac{d_n}{t_n}\right) + \gamma_n^t\left(t_n + \frac{c_n}{h_n^c}\right)\right]\alpha_n \\ &+ \left(\gamma_n^e f_n c_n + \gamma_n^t \frac{c_n}{h_n}\right)(1-\alpha_n) \\ &+ \lambda\left[\left(t_n + \frac{c_n}{h_n^c}\right)\alpha_n + \frac{c_n}{h_n}(1-\alpha_n) - T_{n,\max}\right], \end{aligned} \tag{19}$$

$$\begin{aligned} \frac{\partial L}{\partial \alpha_n^*} &= \gamma_n^e \frac{t_n^*}{g_n^2} h\left(\frac{d_n}{t_n^*}\right) + \gamma_n^t\left(t_n^* + \frac{c_n}{h_n^c}\right) - \left(\gamma_n^e f_n c_n + \gamma_n^t \frac{c_n}{h_n^c}\right) \\ &+ \lambda^*\left(t_n^* + \frac{c_n}{h_n^c} - \frac{c_n}{h_n}\right) = 0, \end{aligned} \tag{20}$$

$$\frac{\partial L}{\partial t_n^*} = \alpha_n^* \frac{\gamma_n^e}{g_n^2} h\left(\frac{d_n}{t_n^*}\right) + \alpha_n^* \frac{\gamma_n^e t_n^*}{g_n^2} h'\left(\frac{d_n}{t_n^*}\right) + \gamma_n^t \alpha_n^* + \lambda^* \alpha_n^* = 0, \tag{21}$$

$$\lambda^*\left[\left(t_n^* + \frac{c_n}{h_n^c}\right)\alpha_n^* + \frac{c_n}{h_n}(1-\alpha_n^*) - T_{n,\max}\right] = 0, \quad \lambda > 0. \tag{22}$$

By denoting $X = (d_n/Bt_n^*)$, and based on equation (21), X satisfies the following:

$$(1 - X\ln 2)2^X = -\frac{g_n^2(\gamma_n^t + \lambda)}{(N_0 + I)B\gamma_n^e} + 1. \tag{23}$$

From equation (23), it can be further derived that

$$2^{(1/\ln 2)\ln 2(X-(1/\ln 2))}\ln 2\left(X - \frac{1}{\ln 2}\right) = \frac{g_n^2(\gamma_n^t + \lambda)}{(N_0 + I)Be\gamma_n^e} - \frac{1}{e}. \tag{24}$$

According to the Lambert W function

$$Q = ze^z \longrightarrow z = W_o(Q), \quad Q \geq -\frac{1}{e}. \tag{25}$$

The inverse function of equation (24) is given as

$$X = \frac{1}{\ln 2}\left[W_0\left(\frac{g_n^2(\gamma_n^t + \lambda)}{(N_0 + I)Be\gamma_n^e} - \frac{1}{e}\right) + 1\right]. \tag{26}$$

Thus, the optimal transmission rate is expressed as

$$r_n^* = \frac{B}{\ln 2}\left[W_0\left(\frac{g_n^2(\gamma_n^t + \lambda)}{(N_0 + I)Be\gamma_n^e} - \frac{1}{e}\right) + 1\right]. \tag{27}$$

For a given $\lambda > 0$, the optimal solution t_n^* and p_n^* of this EEC minimization problem can be calculated as follows:

$$t_n^* = \frac{d_n \ln 2}{B\left[W_0\left(\left(\left(g_n^2(\gamma_n^t + \lambda)\right)/\left((N_0 + I)Be\gamma_n^e\right)\right) - (1/e)\right) + 1\right]}. \tag{28}$$

Meanwhile, according to equations (3) and (4), p_n^* is given by

$$p_n^* = \frac{(N_0 + I)B}{g_n^2}\left(2^{(d_n/Bt_n^*)} - 1\right). \tag{29}$$

The offloading policy can be obtained by solving equation (22):

$$\alpha_n^* = \frac{\left(T_{n,\max}h_n - c_n\right)h_n^c}{t_n^* h_n h_n^c + c_n h_n - c_n h_n^c}. \tag{30}$$

For a given set of $\mathscr{A}$ and $\mathscr{P}$, the Lagrange multiplier is updated by

$$\lambda(k+1) = \left\{\lambda(k) + \theta(k)\left[T_{n,\max} - \frac{c_n}{h_n}(1-\alpha_n) - \left(t_n + \frac{c_n}{h_n^c}\right)\alpha_n\right]\right\}^+, \tag{31}$$

where $k > 0$ is the iteration index and $\theta(k)$ is the positive iteration step size. Then, the updated Lagrange multiplier in equation (31) can be used to update the transmission power allocation in equations (28) and (29) and the offloading policy in equation (30).

Algorithm 1 gives an outline of the proposed algorithm.

5. Performance Evaluation

In this section, we evaluate the performance of the proposed algorithm. The simulation settings were as follows. We first considered that the mobile edge computing scenario has a coverage of 50 m, where $N = 30$ smartphones are distributed in the coverage region randomly [25]. The computing capacity allocated for MD n on the MEC server was set to $h_n^c = 10$ GHz, and the mobile CPU ability h_n is randomly set to $\{0.5, 0.6, \ldots, 1.0\}$ GHz to illustrate the heterogeneous computing capability of the MDs. We set the initial policy weights $\gamma_n^e = \gamma_n^t = 0.5$, which implies that MD n considers both the computation time and energy consumption. The size of the task was uniformly distributed as (0, 20) MB. Without loss of generality, we let c_n be 737.5 cycles/bit [26]. For the wireless access, we set the channel bandwidth $B = 5$ MHz and $N_0 + I = -100$ dBm [27]. By comparing them with the local computing and full offloading methods, we evaluate the proposed partial offloading scheme.

5.1. Comparison of Energy Consumption and Completion Delay

5.1. Comparison of Energy Consumption and Completion Delay. In this subsection, the energy consumption and completion time of the proposed scheme, considering the variance of the task size, are compared with the local computing approach, full offloading approach, and Li's et al. binary offloading approach in [28].

Figure 2 depicts the energy consumption and completion time for the four schemes. As can be seen from Figure 2, the proposed partial offloading and Li's binary offloading scheme are superior to the local computing and full

Require:
ε: an infinitesimal number;
Ensure:
$\{\mathscr{A}, \mathscr{P}\}$: optimal offloading and transmission power allocation policy;
Initialize: c_n, d_n, γ_n^e, γ_n^t, λ, and $\vartheta(t)$ and iteration index $k \longleftarrow 1$;
  Compute $t_{n,\mathrm{loc}}$, $e_{n,\mathrm{loc}}$ by (5) and (6), respectively;
  Compute $Z_{n,\mathrm{loc}}$ by (7);
while $|\lambda(k+1) - \lambda(k)| > \varepsilon$ **do**
  /*Transmit power allocation*/
  Compute t_n by (28);
  $t_n(k+1) = t_n(k)$;
  Compute p_n by (29);
  $p_n(k+1) = p_n(k)$;
  /* Task offloading policy*/
  Compute α_n by (30);
  $\alpha_n(k+1) = \alpha_n(k)$;
  Compute $e_{n,\mathrm{off}}$, $t_{n,\mathrm{off}}$ by (9) and (11), respectively;
  Compute $Z_{n,\mathrm{off}}$ by (13);
  Compute Z_n by (14);
  /* Lagrangian multiplier update */
  Update Lagrangian multiplier $\lambda(k+1)$ by (31);
  $k = k + 1$;
end while

ALGORITHM 1: Iterative energy-efficient computation offloading algorithm for MD n.

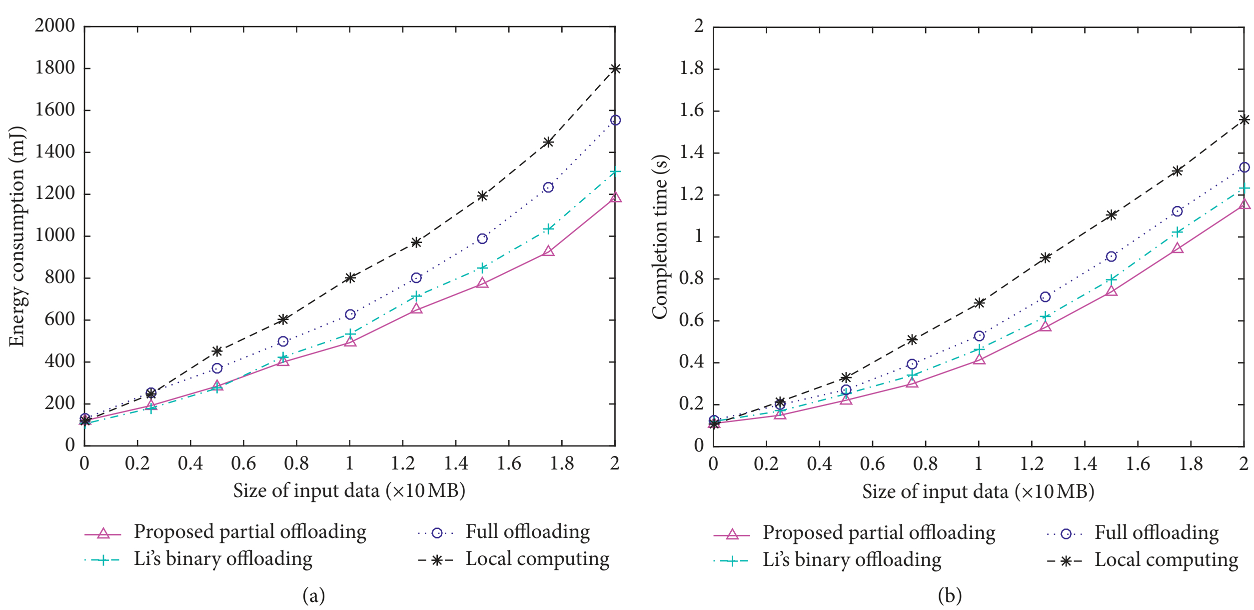

FIGURE 2: Comparison of energy consumption and completion delay for different algorithms. (a) Energy consumption. (b) Completion delay.

offloading schemes. Further, compared with local computing, the full offloading method achieves a better result, especially for greater task sizes. Therefore, for a computing task with a large input data size, the proposed scheme and Li's scheme tend to offload most of the computation tasks to the MEC server to minimize the EEC paid by the MDs. Furthermore, when the size of input data is small, Li's binary offloading approach has the lowest energy consumption. However, this energy consumption increases with the size of input data and eventually is greater than that of the proposed partial offloading scheme. This occurs because the proposed partial offloading scheme not only offloads the computation-intensive subtasks to the MEC server based on the tradeoff between the advantages of the local computing and full offloading, but also exploits the transmission power to reduce energy and time consumption in MEC. In addition, the completion time required for offloading the task to an MEC server includes the communication time in the wireless

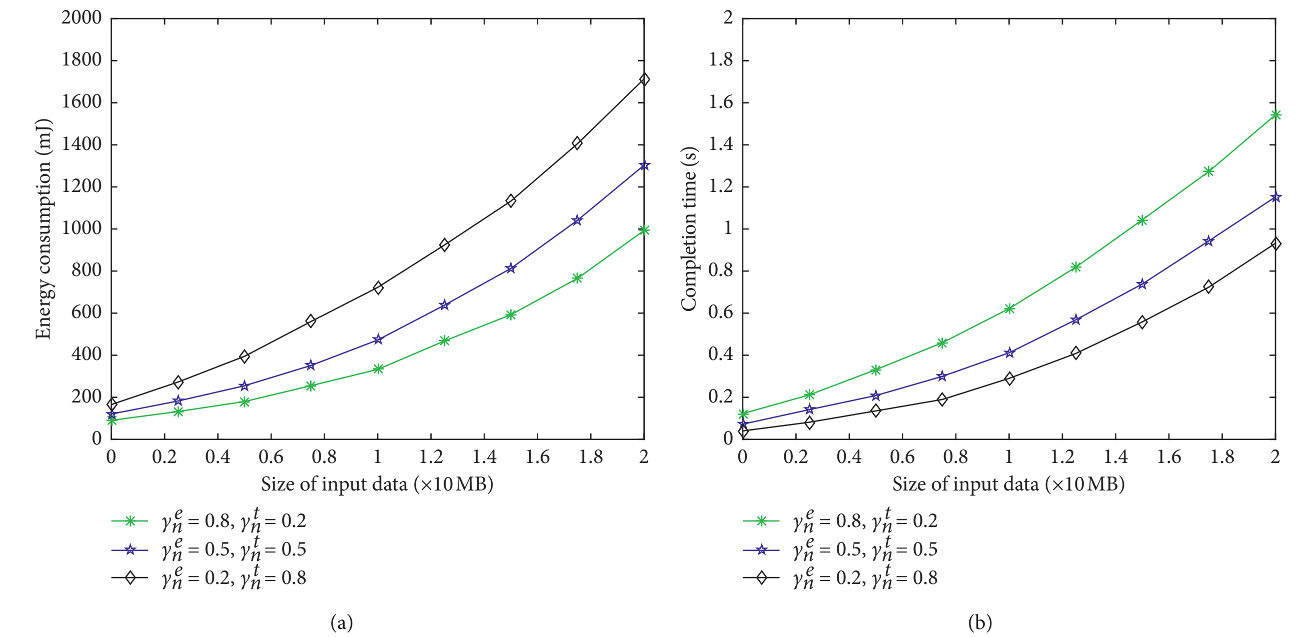

FIGURE 3: Comparison of energy consumption and completion time for different γ_n^e and γ_n^t. (a) Energy consumption. (b) Task completion time.

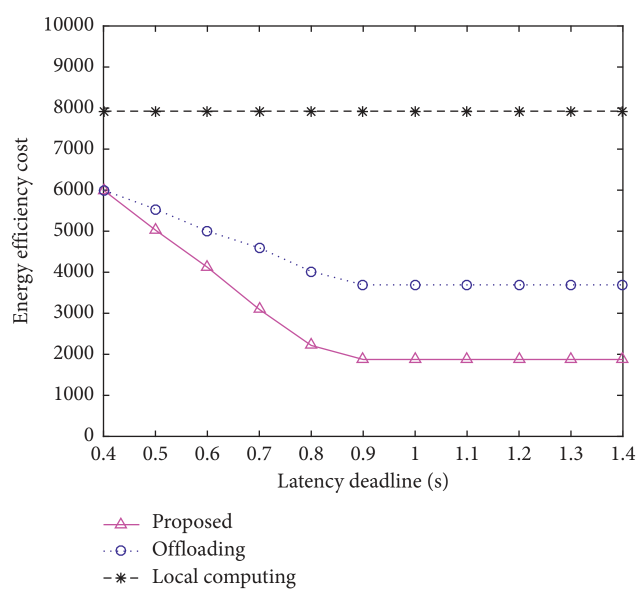

FIGURE 4: Effect of latency requirement on energy efficiency cost (EEC).

network and the execution time in the MEC server. It can be seen in Figure 2(b) that the task completion time of our partial offloading scheme increases relatively slowly with the task size.

5.2. Impact of Weights γ_n^e and γ_n^t. In this subsection, the impact of weights γ_n^e and γ_n^t on the energy consumption and computation time of tasks with different sizes of input data is studied.

Figure 3 shows the differences in energy consumption and computation delay under different settings of γ_n^e and γ_n^t. The energy consumption is observed to increase as γ_n^e decreases, independent of the size of the task; nevertheless, the changes in the computation delay are reversed. This is expected because a large γ_n^e will induce a decrease in transmission rate, as expressed in equation (26), thus causing the decline of transmission power during the edge computing execution.

5.3. Comparison of EEC on Execution Strategy. In this subsection, the proposed task execution algorithm is compared with the other two execution methods, namely, local computing and full offloading, under the hard-completion time deadline constraint.

As shown in Figure 4, the differences in latency requirements affect the EEC for the same task profile. The following observations can be drawn from Figure 4. First, only the task data size and CPU computing capability can affect the EEC of the local computing method. Therefore, the changes in latency requirement do not influence its EEC. Second, compared to local computing, the partial offloading scheme can reduce the EEC significantly. This is because the proposed algorithm can optimally offload a fraction of the computation for execution on the MEC server according to the EEC on the edge and the local device. Third, the proposed algorithm has a lower EEC for a long completion time deadline compared with the full offloading method. This is reasonable given that the proposed algorithm adopts the optimal offloading policy and transmission power allocation.

6. Conclusions

In this work, the problem of energy-efficient computation offloading for mobile edge computing was studied. We integrated computation offloading with transmit power allocation to minimize both energy and time consumption under a completion time deadline constraint. We designed a novel algorithm comprising computation offloading policy and transmission power allocation subalgorithms. The experimental results demonstrated that our proposed method can effectively reduce the energy consumption and completion time by taking advantage of the dynamic computation offloading policy and transmission power allocation for mobile edge computing.

Future work will consider a more general case in which mobile users may depart and leave dynamically during a computation offloading period. In this case, the user migration patterns will greatly affect the problem formulation.

Data Availability

The Matlab simulation results used to support the findings of this study are included within the supplementary information file (available here).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant nos. 61971245 and 61801249. The authors would like to thank Editage (http://www.editage.com) for English language editing.

Supplementary Materials

Mat lab simulation results used to support the findings of this study. (*Supplementary Materials*)

References

[1] K. Kumar and Y. Lu, "Cloud computing for mobile users: can offloading computation save energy?," *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[2] M. A. Ayu, S. A. Ismail, T. Mantoro, and A. F. A. Matin, "Real-time activity recognition in mobile phones based on its accelerometer data," in *Proceedings of the International Conference on Informatics & Computing*, pp. 292–297, Mataram, Indonesia, November 2016.

[3] E. Cuervo, A. Balasubramanian, D. Cho et al., "MAUI: making smartphones last longer with code offload," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services—MobiSys '10*, pp. 49–60, San Francisco, CA, USA, June 2010.

[4] M. Velický and P. S. Toth, "From two-dimensional materials to their heterostructures: an electrochemist's perspective," *Applied Materials Today*, vol. 8, pp. 68–103, 2017.

[5] M. Chen, Y. Qian, Y. Hao, Y. Li, and J. Song, "Data-driven computing and caching in 5G networks: architecture and delay analysis," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 70–75, 2018.

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

[7] S. Yu, R. Langar, X. Fu, L. Wang, and Z. Han, "Computation offloading with data caching enhancement for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11098–11112, 2018.

[8] White Paper, ETSI, *Mobile-Edge Computing—Introductory Technical White Paper*, White Paper, ETSI, Sophia Antipolis, France, 2014, https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf.

[9] L. Liu, Z. Chang, X. Guo, and T. Ristaniemi, "Multi-objective optimization for computation offloading in mobile-edge computing," in *Proceedings of the Computers & Communications*, pp. 832–837, Heraklion, Greece, July 2017.

[10] Q. Hong, A. S. Hussein, and K. Anja, "Efficient resource allocation in mobile-edge computation offloading: completion time minimization," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pp. 2513–2517, Aachen, Germany, June 2017.

[11] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pp. 1451–1455, Barcelona, Spain, July 2016.

[12] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[13] 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.

[14] K. Zhang, Y. Mao, S. Leng et al., "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[15] C. You, Y. Zeng, R. Zhang, and K. Huang, "Asynchronous mobile-edge computation offloading: energy-efficient resource management," *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7590–7605, 2018.

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

[17] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, San Francisco, CA, USA, March 2017.

[18] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[19] C. You and K. Huang, "Multiuser resource allocation for mobile-edge computation offloading," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Washington, DC, USA, December 2016.

[20] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 319–333, 2019.

[21] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," *IEEE Wireless Communications Letters*, vol. 6, no. 6, pp. 774–777, 2017.

[22] G. Gao, M. Xiao, J. Wu, K. Han, L. Huang, and Z. Zhao, "Opportunistic mobile data offloading with deadline constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 12, pp. 3584–3599, 2017.

[23] X. Meng, W. Wang, and Z. Zhang, "Delay-constrained hybrid computation offloading with cloud and fog computing," *IEEE Access*, vol. 5, pp. 21355–21367, 2017.

[24] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, 2018.

[25] T. Q. Quek, G. de la Roche, I. Güvenç, and M. Kountouris, *Small Cell Networks: Deployment, PHY Techniques, and Resource Management*, Cambridge Univ. Press, Cambridge, UK, 2013.

[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] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Upper Saddle River, NJ, USA, 1996.

[28] L. Li, X. Zhang, K. Liu, and F. Jiang, "An energy-aware task offloading mechanism in multiuser mobile-edge cloud computing," *Mobile Information Systems*, vol. 2018, Article ID 7646705, 12 pages, 2018.