



Article

# Jointly Optimize the Residual Energy of Multiple Mobile Devices in the MEC–WPT System

Long Li <sup>1,\*</sup>, Gaochao Xu <sup>1</sup>, Peng Liu <sup>2,\*</sup>, Yang Li <sup>3</sup> and Jiaqi Ge <sup>1</sup>

<sup>1</sup> College of Computer Science and Technology, Jilin University, Changchun 130012, China; xugc@jlu.edu.cn (G.X.); gejq18@mails.jlu.edu.cn (J.G.)

<sup>2</sup> College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China

<sup>3</sup> School of Information Science and Technology, North China University of Technology, Beijing 100144, China; liyang17@mails.jlu.edu.cn

\* Correspondence: lilong18@mails.jlu.edu.cn (L.L.); pliu16@mails.jlu.edu.cn (P.L.)

Received: 10 November 2020; Accepted: 17 December 2020; Published: 20 December 2020



**Abstract:** With the rapid popularity of mobile devices (MDs), mobile edge computing (MEC) networks and wireless power transmission (WPT) will receive more attention. Naturally, by integrating these two technologies, the inherent energy consumption during task execution can be effectively reduced, and the collected energy can be provided to charge the MD. In this article, our research focuses on extending the battery time of MDs by maximizing the harvested energy and minimizing the consumed energy in the MEC–WPT system, which is formulated as a residual energy maximization problem and also a non-convex optimization problem. On the basis of study on maximizing the residual energy under multi-users and multi-time blocks, we propose an effective jointly optimization method (i.e., jointly optimize the energy harvesting time, task-offloading time, task-offloading size and the MDs' CPU frequency), which combines the convex optimization method and the augmented Lagrangian to solve the residual energy maximum problem. We leverage Time Division Multiple Access (TDMA) mode to coordinate computation offloading. Simulation results show that our scheme has better performance than the benchmark schemes on maximizing residual energy. In particular, our proposed scheme is outstanding in the failure rate of multiple MDs and can adapt to the task size to minimize the failure rate.

**Keywords:** mobile edge computing; wireless power transmission; convex optimization; augmented Lagrangian method

## 1. Introduction

With the explosive development of the mobile devices and mobile communication, mobile devices (MDs) are becoming more and more widely used, the novel mobile applications emerge continuously. These emerging latency-sensitive applications rely heavily on real-time communication and computation of wireless devices [1]. However, most MDs have low computing capability and finite battery capacity, and cannot meet the massive demand for physical resources of current applications, which will be a crucial and breakthrough challenge. To resolve this issue, mobile edge computing (MEC) has emerged as a promising technique by providing cloud-like computing at the edge of mobile networks via integrated MEC servers at wireless access points (APs) [2]. Leveraging MEC, MDs with limited resources can offload their computation tasks to APs, and remotely compute and offload tasks on the MEC server. Generally, there are two different methods for computation, namely binary offloading and partial offloading. When using binary offloading, the division of tasks is not allowed. The entire task is required to be executed locally on the MD or remotely on the edge server. As for partial offloading, the task can be divided into two parts, one part is executed locally and the other

part is executed on the edge server [3]. Meanwhile, because MEC technology can promote low-power devices to perform computing tasks in real-time, and MEC technology has attracted growing research interests in both academia and industry.

In addition to improving the computation capability and reducing power consumption of MDs, another challenge is how to provide a sustainable and cost-effective energy supply to massive computation-heavy MDs is another challenge. The development of radio frequency (RF)-based wireless power transmission (WPT) technology brings a new solution to solve the problem of insufficient battery energy [4]. It was shown that user quality of experience (QoE) can be improved by integrating wireless powered techniques into MEC networks since the duration of having MEC services is extended [5]. By integrating WPT with the AP, the dedicated energy transmitter can be deployed to spread energy wirelessly to continuously power the battery of remote energy harvesting (EH) devices [6]. Deploying multiple antennas on the energy transmitter can be used for multiple devices at the same time, which helps to improve the efficiency of WPT, thereby it can satisfy more MDs' energy supply requests.

In this article, we study the integration of WPT into MEC where multiple MDs collect energy in each time block and perform computationally intensive tasks simultaneously on edge servers and local devices. The key contributions of this article are as follows.

- (1) We adopting the case of multiple MDs, leveraging multiple antennas on the AP to perform energy transmission and charging multiple MDs at the same time so that each MD completes the corresponding computing task based on the harvested energy. Using partial offloading, each MD arbitrarily divides the computing task into two independent parts, which are used for local computing and task offloading. In addition, our proposed scheme uses Time Division Multiple Access (TDMA) protocol to coordinate computation offloading, where different users offload their respective tasks to the AP over orthogonal time slots.
- (2) In the integrated system, we focus on the joint optimization of the communication and computing resources of the partial computation offloading system. Taking the multi-period execution tasks of MDs into account, we achieve the goal of maximizing the remaining energy under multiple MDs and multi-slots, which requires maximizing energy harvesting and minimizing energy consumption. Therefore, we jointly optimize the task offloading size, task offloading time, energy harvesting time and CPU frequency of the MDs.
- (3) We established the system model and formalized the residual energy maximum problem. The best solution was obtained by proposing effective convex optimization methods and the augmented Lagrangian method. The simulation results show that the proposed joint optimizing scheme outperforms the previous ones in the partial computation offloading system, and the proposed algorithm is proven to be more efficient.

The remainder of the article is organized as follows. The related work is introduced in Section 2. In Section 3, we describe the WPT and MEC integrated system model in details. Section 4 gives the formulation and analysis of the problem. In Section 5, we provide the implemented algorithm. We give and discuss the experimental results in Section 6. Finally, we conclude the results in this article.

## 2. Related Work

In recent years, with the development of mobile edge computing and mobile cloud computing, computation offloading attracts more attention. The development of computational offloading can effectively improve the energy efficiency and task processing rate of MDs. Reducing the power consumption of MDs and increasing the working hours of the battery is a great concern to users. In previous studies, it was proved that offloading to the cloud can save energy for MDs and extend battery standby time [7], and refs. [8–10] proposed different computation offloading frameworks. Therefore, there are two basic computing task offloading models in MEC, namely binary offloading and partial offloading [3]. The combination of MEC networks promotes the execution of a large number

of computing tasks on low-power devices, arousing the great attention of the resource management of computing offloading in the academic community [11]. Specifically, the multi-user mobile-edge computation offloading (MECO) system is studied in [12,13], and in [12], the mobile device delay minimization resource allocation of the MECO system under multiple access is mainly considered. In [14], the resource allocation in the asynchronous MECO system with heterogeneous input data arrival time and calculation deadline is studied. Of course, there are also related researches on resource scheduling strategies on edge servers. In [15], a two-level cooperative resource allocation mechanism is proposed, which provides formal guarantees regarding resource allocation and QoS specifications at the lower level, and a way to solve the problem of edge server workload balancing at the upper level.

Computation offloading can effectively improve the task processing capabilities of MDs, which largely depends on the quality of communication because data transmission is essential. Combining previous research, we conclude that through the joint optimization of communication and computing resources [16], the obtainable is significantly better than the single optimization method, which prompts us to focus on joint optimization. Leveraging Augmented Reality (AR) property in [17], different user applications share part of the computational tasks and of the input and output data, to reduce communication and computing overhead via the joint optimization of communication resources and computing resources. In [18], a joint optimization of the transmission power, the number of bits per symbol and the CPU cycles assigned to each application was proposed to minimize the power consumption of the mobile terminal. Wang et al. considered the joint optimization of communication, computational offloading, resource allocation and content caching, and transformed the original problem to the convex problem [19]. The distributed convex optimization method and the algorithm based on the Alternating Direction Multiplier Method (ADMM) are used to solve the optimization problem. In [20], the (TDMA) communication protocol is adopted, which jointly optimizes task allocation, time and power allocation, combined with the performance limitations of local and MEC edge servers to minimize the computation delay of local users. This is a mixed-integer nonlinear program (MINLP), which is relaxed to a convex problem and is solved by a sub-optimal solution based on the relaxed convex problem. The authors in [21] proposed a multi-layer data stream processing system, where the cloud server is in the top layer, the mobile edge computing server is in the middle layer and the edge devices are at the bottom layer, and the system offloads tasks and computing resources through multi-layer processing. However, the above methods are only expected to reduce energy consumption to achieve the goal of energy saving. Therefore, some studies propose the idea of combining WPT technology to charge MDs [22]. They apply energy harvesting to computation offloading and local computing to reduce energy consumption while effectively helping MDs to charge. The above methods are similar to ours, but we propose the joint optimization method to minimize system delay in a multi-layer system.

Combined with the latest research findings, researchers began to focus on the close integration of WPT and MEC [23–26]. The wireless multi-user MEC system composed of multi-antenna AP and multiple single-antenna users is studied, the downlink WPT and uplink wireless communication (used to computation offloading) are performed simultaneously on orthogonal frequency bands. Through joint optimization of task-offloading time, AP energy transmission beam formation, task-offloading size and CPU frequency, while taking into account the constraints on task completion time and user energy harvesting, the convex optimization problem is obtained. By leveraging the state-of-the-art optimization techniques, the optimal solution to minimizing the overall energy consumption of the AP is obtained in a semi-closed form [22]. In [27], it is emphasized the MEC–WPT system of two mobile devices, and the two devices are at a far point and a near point, respectively. Fully considering the dual near-far effects of the distant mobile devices, the closer mobile devices cooperate in the form of the relay to perform offloading, and then according to the constraints of task-offloading to minimize the total AP emission energy. Finally, an effective problem form is obtained, and the two-stage method is used to solve the problem to get the best solution. The MEC system using multiple MDs and combined with WPT technology is studied, considering the goal of maximizing the

remaining energy of MDs while ensuring more efficient energy collection efficiency and lower energy consumption. In the process of computation task and energy harvesting, the parameters are jointly optimized to maximize the remaining energy of the system and propose corresponding solutions to obtain the best solution [28]. The authors in [13,29] extended the energy consumption minimization problems into multi-user MEC networks with TDMA and orthogonal frequency-division multiple access, respectively. Energy-efficient TDMA and non-orthogonal multiple access (NOMA) using partial offloading and binary offloading are studied in the wireless-powered MEC networks [30].

Regarding the previous research, most of the research focused on reducing energy consumption, which is a way of thinking for service providers rather than users. Only a few studies think about problems from the user’s point of view, studying the remaining energy of MDs to ensure the user’s service quality. Among them, most papers consider only one user in a time block is studied, but not the multi-user situation. The purpose of this article was to maximize the remaining energy in multiple time slots by studying multiple MDs in the case of time-division multiplexing and to ensure that the overall energy consumption is the minimum and the residual energy is the maximum.

### 3. System Model

This article considers a wireless powered multi-user MEC system, which consists of an multi-antenna AP and a set  $M \triangleq \{1, \dots, M\}$  of single-antenna MDs. The AP is directly integrated in the MEC server and has  $N \triangleq \{1, \dots, N\}$  time slots. The AP not only receives task data and transmits the data to the edge server, but also powers MDs through WPT. MDs can offload and execute computation tasks, and harvest energy from APs. However, since the MD is a single antenna, the MD cannot offload tasks and harvest energy at the same time. The system model is shown in Figure 1.

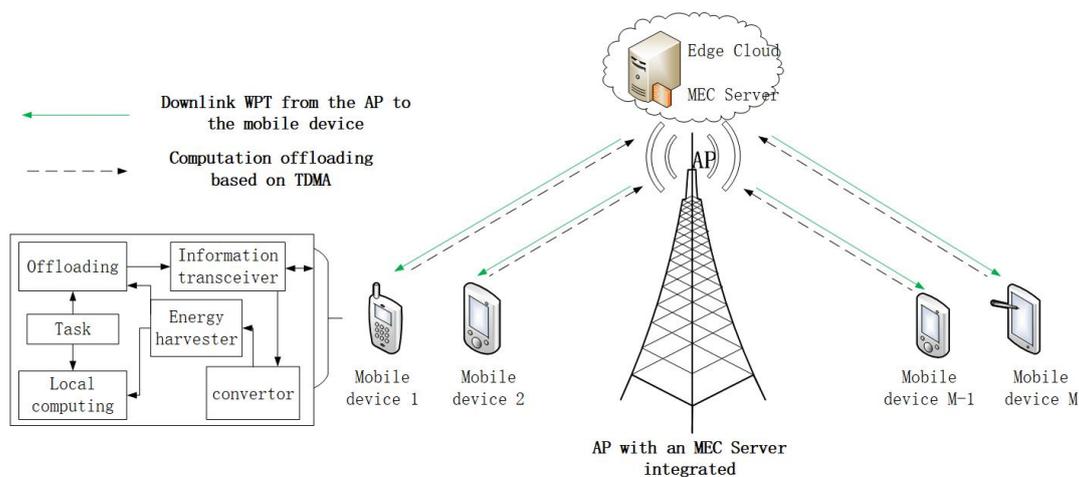


Figure 1. System model.

Taking the communication requirements of MDs during task offloading into account, we adopt a slot-based time division multiplexing (TDMA) model. We assume that the length of each time block is  $T$  and the channel remains static in one time block but changes between blocks. The MD collects energy at the beginning of each time block and then offloads the computing task to the edge server during the remaining time. The computing task is divided into two parts and executed on the local and edge servers, respectively. As for the local computation part, the execution starts at the beginning and will not stop until use up the whole time block. In order to understand the AP’s energy transmission and computing offloading better, we assume that the AP has perfect knowledge of the channels from AP to MDs and the computation requirement for each task, and it can coordinate the task partition, computation offloading and local computing by jointly allocating both communication and computation resources. The parameters are given in Table 1.

Table 1. Parameters .

Parameters	Notation
The input bit of the task	$R_{i,j}$
The offload bit	$R_{i,j}^{off}$
The number of cycles for one bit	$C_j$
The effective capacitance coefficient	$\kappa_j$
The frequency of the CPU	$f_{i,j}$
The communication bandwidth	$B$
The channel gain from MD to AP	$h_{i,j}^m$
The transmission power from MD to AP	$P_{i,j}^m$
The distance between device and AP	$d_{i,j}$
The constant circuit power	$P_{m,j}$
Energy conversion efficiency	$\nu_j$
The channel gain of downlink channel	$h_{i,j}^a$
The AP transmission power	$P_{i,j}^a$
The energy harvesting time	$t_i^{har}$

### 3.1. Local Computing Model

For the task,  $R_{i,j}$  and  $R_{i,j}^{off}$  denote the input bit and offload bit of the tasks, respectively.  $j \in M$ ,  $i \in N$ . As for the local computing, the size of the local computing data on the  $j$ -th MD after knowing the task size and offloading size is  $R_{i,j}^{loc} = R_{i,j} - R_{i,j}^{off}$ . We can get the local computing energy consumption as:

$$E_{i,j}^{loc} = (R_{i,j} - R_{i,j}^{off})C_j P_{i,j} \quad i \in N, j \in M \tag{1}$$

where  $P_{i,j} = \kappa_j f_{i,j}^2$  is the energy consumption per cycle,  $C_j$  denotes the cycles required to process one bit input, which depends on the task characteristic,  $\kappa_j$  is the effective capacitance coefficient, depending on the MD chip architecture of user  $j$ ,  $f_{i,j}$  denotes the frequency of the CPU of the  $j$ -th MD in  $i$ -th time slot, and is constrained by  $f_{i,j} \in (0, f_j^{max})$ .

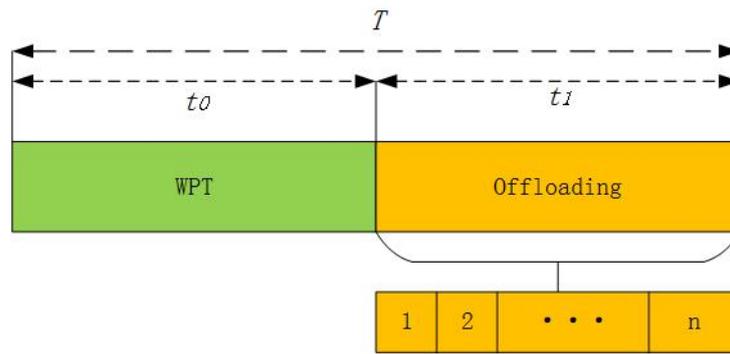
According to the CPU frequency, we can get the time consumed by the local computation as:

$$T_{i,j}^{loc} = \frac{(R_{i,j} - R_{i,j}^{off})C_j}{f_{i,j}} \quad i \in N, j \in M \tag{2}$$

After obtaining the time consumed by the local computation, according to the previous assumption, we use (2) as a constraint to perform the computation task on the MD. The user cannot stand for waiting for a long time, so the time limit of the local computation should be within the time block. Thus, we assume that  $T_{i,j}^{loc} \leq T$ .

### 3.2. Task-Offloading Model

Considering that energy harvesting and task offloading are performed by multiple MDs in a time block, we use the TDMA protocol, which is shown in Figure 2.



**Figure 2.** Time-division multiplexing model for energy harvesting and task offloading.

In Figure 2,  $t_0$  represents the energy harvesting time of all MDs in the time block. Multiple MDs parallel the energy harvesting, so the energy harvesting time of each MD in the same time slice is equal.  $t_1$  denotes the task-offloading time of all MDs in the time block, and there is no requirement that  $t_0$  and  $t_1$  must be equal. Here, we suppose that if there is a multi-core CPU perfect well on the edge server, the execution time of the computing task offloaded to the edge server can be ignored. Compared with the size of the offloading task, the size of the computation result is usually much smaller than the offloaded task and can be downloaded directly by the MD, so we also ignore the download time. According to the above assumptions, we can clearly know that the task-offloading model is mainly about offloading communication.

From the Shannon formula in communication knowledge, we know that the transmission speed of offloading is:

$$r_{i,j} = B \log_2 \left( 1 + \frac{p_{i,j}^m h_{i,j}^m}{N_0 d_{i,j}} \right) \quad i \in N, j \in M \tag{3}$$

where  $h_{i,j}^m$  represents the channel gain from MD  $j$  to AP, which is a constant during the offload duration,  $p_{i,j}^m$  represents the transmission power from MD  $j$  to the AP,  $N_0$  represents the noise power from device  $j$  to the AP,  $d_{i,j}$  indicates the distance between device  $j$  and AP.  $R_{i,j}^{off}$  is the offloading data size, where  $r_{i,j} = \frac{R_{i,j}^{off}}{t_{i,j}^{off}}$ . For simple expression, we introduce a function  $f(x) = N_0 d_{i,j} (2^{\frac{x}{B}} - 1)$ , then the

transmission power of offloading from MD  $j$  to the AP is  $p_{i,j}^m = \frac{1}{h_{i,j}^m} f\left(\frac{R_{i,j}^{off}}{t_{i,j}^{off}}\right)$ . Note that to avoid the

problem of division by zero, we define  $f\left(\frac{R_{i,j}^{off}}{t_{i,j}^{off}}\right) = 0$  when  $R_{i,j}^{off}$  or  $t_{i,j}^{off} = 0$ . In addition, a constant circuit power  $p_{m,j}$  (such as digital-to-analog converter (DAC), filter, etc.) is consumed. We can get the communication energy consumption of the task offload model:

$$E_{i,j}^{off} = \frac{t_{i,j}^{off}}{h_{i,j}^m} f\left(\frac{R_{i,j}^{off}}{t_{i,j}^{off}}\right) + p_{m,j} t_{i,j}^{off} \quad i \in N, j \in M \tag{4}$$

We consider that the computing power of the MEC server is limited and can help us obtain more realistic results. We suppose  $F$  represents the maximum computing power of the MEC server in each time block. Here we use the maximum number of CPU cycles in a time block to represent:

$$\sum_{j=1}^M R_{i,j}^{off} C_j \leq F \quad i = 1, 2, \dots, N \tag{5}$$

This constraint ensures that the computing delay of the edge server is negligible.

### 3.3. Energy Harvesting Model

The multi-antenna AP in the multi-user MEC system can use the energy beamforming to transfer power to the MD through WPT technology, and the single antenna of the MD can receive the energy transmitted by the AP to charge its battery. Although the current energy harvesting technology is not very mature, we consider the relatively sufficient battery power. Therefore, we assume that MDs have sufficient battery capacity to store harvested energy. Then the energy collected by the MD during energy harvesting is:

$$E_{i,j} = v_j p_{i,j}^a h_{i,j}^a t_i^{har} \quad i \in N, j \in M \quad (6)$$

where  $0 < v_j < 1$  represents energy conversion efficiency,  $h_{i,j}^a$  represents the channel gain of the downlink channel from the AP to the  $j$ -th MD and is a constant during the harvest time,  $p_{i,j}^a$  represents the AP transmission power,  $t_i^{har}$  represents the energy harvesting time of the MD in the  $i$ -th time block and the energy harvesting time of the MDs in this time block is equal.

According to our model, the MD harvests energy first, and then offloads the task during the rest of the block to meet the time constraint of time  $T$  in each time block. In our system, multiple MDs perform energy harvesting and task-offloading in each time block. Therefore, the energy harvesting time and task-offloading time must meet the following constraints:

$$t_i^{har} + \sum_{j=1}^M t_{i,j}^{off} C_j \leq T \quad i \in N \quad (7)$$

The above are the three aspects of the system model for which we have carried out a detailed description and explanation.

## 4. Formalization of the Problem

The goal of this article was to maximize the residual energy of multiple MDs in multiple time blocks, which requires to maximize the harvested energy and minimize the consumed energy at the same time. The amount of harvest energy is directly proportional to the energy harvest time  $t_i^{har}$ , larger  $t_i^{har}$  will surely harvest more energy but also minimize the maximum available time for task offloading, which brings high energy consumption and the risk of violating time constraints. With each given  $t_i^{har}$ , we should coordinate the task-offloading size  $R_{i,j}^{off}$ , the task-offloading time  $t_{i,j}^{off}$  and the CPU frequency of local execution to minimize the energy consumption. In short, in order to maximize the residual energy of MDs, we need to jointly optimize the energy harvesting time, task-offloading size, task-offloading time and CPU frequency to find the optimal combination. The following is the formulation to describe the residual energy maximization problem under multiple MDs and multiple time slots.

### 4.1. Problem Formulation

According to the proposed model, the residual energy of each MD in any time block is expressed as a function as:

$$E_{i,j}^{res}(f_{i,j}, R_{i,j}^{off}, t_i^{har}, t_{i,j}^{off}) = E_{i,j} - (E_{i,j}^{loc} + E_{i,j}^{off}) \quad (8)$$

where the  $E_{i,j}$ ,  $E_{i,j}^{off}$ ,  $E_{i,j}^{loc}$  are given by (6), (4) and (1), respectively.

Therefore, the problem of maximizing residual energy in multi-user and multi-time slots can be expressed as:

$$\begin{aligned}
 P1 \quad & \max_{f, R^{off}, t^{har}, t^{off}} \sum_{i=1}^N \sum_{j=1}^M E_{i,j}^{res}(f_{i,j}, R_{i,j}^{off}, t_i^{har}, t_{i,j}^{off}) \\
 \text{s.t.} \quad & C1: \frac{(R_{i,j} - R_{i,j}^{off})}{f_{i,j}} \leq T \quad i \in N, j \in M \\
 & C2: 0 < f_{i,j} \leq f_{i,j}^{max} \quad i \in N, j \in M \\
 & C3: t_i^{har} + \sum_{j=1}^M t_{i,j}^{off} \leq T \quad i \in N \\
 & C4: \sum_{j=1}^M R_{i,j}^{off} C_j \leq F \quad i \in N \\
 & C5: t_i^{har} \geq 0 \quad i = 1, 2, 3, \dots, N \\
 & C6: t_{i,j}^{off} \geq 0 \quad i = 1, 2, 3, \dots, N, j \in M \\
 & C7: 0 < R_{i,j}^{off} \leq R_{i,j} \quad i = 1, 2, 3, \dots, N, j \in M \\
 & C8: E_{i,j} - (E_{i,j}^{loc} + E_{i,j}^{off}) \geq 0 \quad i \in N, j \in M
 \end{aligned} \tag{9}$$

In order to simplify the formula, we define  $f(i, j) = v_j p_{i,j}^a h_{i,j}^a$ , then the complete expression of:

$$E_{i,j}^{res}(f_{i,j}, R_{i,j}^{off}, t_i^{har}, t_{i,j}^{off}) = f(i, j) t_i^{har} - ((R_{i,j} - R_{i,j}^{off}) C_j \kappa_j f_{i,j}^2 + \frac{t_{i,j}^{off}}{h_{i,j}^m} f(\frac{R_{i,j}^{off}}{t_{i,j}^{off}}) + p_{m,j} t_{i,j}^{off}) \tag{10}$$

In P1, C1 and C3 respectively represent the time constraints of local computation and edge offloading, the total time of offloading time and energy harvesting time cannot be greater than each time block  $T$ , and local computation can use the entire time block  $T$ . C2 is the CPU frequency constraint of each MD, and C4 is the computing power constraint of the edge server. C5, C6 and C7 give the optimization variable range of the problem. C8 ensures the residual energy of each MD in each time block.

The optimal solution of problem P1 contains four solution vectors, where  $f_{i,j} = [[f_{1,1}, f_{1,2}, \dots, f_{1,M}], [f_{2,1}, f_{2,2}, \dots, f_{2,M}], \dots, [f_{N,1}, f_{N,2}, \dots, f_{N,M}]]$  consists of the CPU frequency of each MD,  $R_{i,j}^{off} = [[R_{1,1}^{off}, R_{1,2}^{off}, \dots, R_{1,M}^{off}], [R_{2,1}^{off}, R_{2,2}^{off}, \dots, R_{2,M}^{off}], \dots, [R_{N,1}^{off}, R_{N,2}^{off}, \dots, R_{N,M}^{off}]]$  consists of the size of each MD's offload data,  $t_i^{har} = [t_1^{har}, t_2^{har}, \dots, t_N^{har}]$  consists of the energy harvesting time of the MD in each time block,  $t_{i,j}^{off} = [[t_{1,1}^{off}, t_{1,2}^{off}, \dots, t_{1,M}^{off}], [t_{2,1}^{off}, t_{2,2}^{off}, \dots, t_{2,M}^{off}], \dots, [t_{N,1}^{off}, t_{N,2}^{off}, \dots, t_{N,M}^{off}]]$  consists of task offload time for each MD.

#### 4.2. Problem Analysis

Problem P1 is a non-convex problem that is difficult to solve. However, through the analysis of problem P1, we can first transform the main problem into a problem of smaller dimensions. Here we introduce Lemma 1, which is the basis of transformation.

**Lemma 1.** We always have  $\inf_{x,y} f(x, y) = \inf_x \tilde{f}(x)$ , where  $\inf_x \tilde{f}(x) = \inf_y f(x, y)$ .

**Proof of Lemma 1.** See [31]. □

Lemma 1 shows that the objective function can be minimized by first minimizing some of variables, and then minimizing the other ones. For this reason, we are able to find the optimal  $f$  firstly.

According to Equation (3), we can find that the energy consumption of local computation increases monotonically with the increase of  $T_1$ , the smaller the  $T_1$  is, the less energy consumed. However, the small  $f_{i,j}$  lead to the violation of constraint C2. In other words, local task execution cannot be completed within time block  $T$ . Therefore, when the time  $T$  is taken as the maximum available computation time, the best value  $f_{i,j}^* = \frac{(R_{i,j} - R_{i,j}^{off})C_j}{T}$ .

It can be seen that the objective function and  $f(\frac{R_{i,j}^{off}}{t_{i,j}^{off}}) = 0$  satisfy the constraint 6 so that the constraint 6 can be eliminated. When all tasks are offloaded to the edge server,  $R_{i,j} - R_{i,j}^{off} = 0$ , which means that no local computation is required. At the same time,  $f_{i,j}^* = 0$  satisfies constraint 7. By replacing  $f_{i,j}^*$  in  $P1$ , we can simplify the original problem  $P1$  to  $P2$ :

$$\begin{aligned}
 P2 \quad & \max_{R^{off}, t^{har}, t^{off}} \sum_{i=1}^N \sum_{j=1}^M E_{i,j}^{res}(R_{i,j}^{off}, t_i^{har}, t_{i,j}^{off}) \\
 \text{s.t.} \quad & C3, C4, C5, C7, C8
 \end{aligned} \tag{11}$$

where  $E_{i,j}^{res}(R_{i,j}^{off}, t_i^{har}, t_{i,j}^{off}) = f(i, j)t_i^{har} - (\frac{\kappa_j((R_{i,j} - R_{i,j}^{off})C_j)^3}{T^2} + \frac{t_{i,j}^{off}}{h_{i,j}^m} f(\frac{R_{i,j}^{off}}{t_{i,j}^{off}}) + p_{m,j}t_{i,j}^{off})$ , C1, C2 and C6 can be eliminated.

Through the above replacement steps and constraint elimination, the problem  $P1$  is converted to  $P2$ , and a variable is eliminated simultaneously. Next, we will solve the problem  $P2$  to get the best combination of offload data size, energy harvesting time and task offload time.

### 5. Problem Solving

Through the above simplification, we have obtained the simplification problem  $P2$ . In order to get the best task-offloading time and energy harvesting time, we have the constraint  $t_i^{har} + \sum_{j=1}^M t_{i,j}^{off} \leq T$ . After the time  $T$  is consumed, the maximum energy harvesting time and the minimum task-offloading time are the best solutions. Therefore, let  $t_i^{har} + \sum_{j=1}^M t_{i,j}^{off} = T$ , then  $t_i^{har} = T - \sum_{j=1}^M t_{i,j}^{off}$ . After we replace  $t_i^{har}$ , we get the  $P3$  question:

$$\begin{aligned}
 P3 \quad & \max_{R^{off}, t^{off}} \sum_{i=1}^N \sum_{j=1}^M E_{i,j}^{res}(R_{i,j}^{off}, t_{i,j}^{off}) \\
 \text{s.t.} \quad & C4, C7, C8 \\
 & C9: \sum_{j=1}^M t_{i,j}^{off} < T \quad i \in N \\
 & C10: t_{i,j}^{off} > 0 \quad i = 1, 2, \dots, N, j \in M
 \end{aligned} \tag{12}$$

where  $E_{i,j}^{res}(R_{i,j}^{off}, t_{i,j}^{off}) = f(i, j)T - (f(i, j) \sum_{j=1}^M t_{i,j}^{off} + \frac{\kappa_j((R_{i,j} - R_{i,j}^{off})C_j)^3}{T^2} + \frac{t_{i,j}^{off}}{h_{i,j}^m} f(\frac{R_{i,j}^{off}}{t_{i,j}^{off}}) + p_{m,j}t_{i,j}^{off})$ , and the C3 and C5 are replaced by C9 and C10.

Through the above simplification, the problem is divided into two parts, one is a constant term and the other is a non-constant term with variables. It happens that the constant term subtracts the non-constant term, then we transform the P3 problem into the P4 problem:

$$\begin{aligned}
 P4 \quad & \min_{R^{off}, t^{off}} \sum_{i=1}^N \sum_{j=1}^M E_{i,j}^{con}(R_{i,j}^{off}, t_{i,j}^{off}) \\
 \text{s.t.} \quad & C1: \sum_{j=1}^M R_{i,j}^{off} C_j \leq F \quad i \in N \\
 & C2: 0 < R_{i,j}^{off} \leq R_{i,j} \quad i \in N, j \in M \\
 & C3: (E_{i,j}^{loc} + E_{i,j}^{off}) - E_{i,j} \leq 0 \quad i \in N, j \in M \\
 & C4: \sum_{j=1}^M t_{i,j}^{off} < T \quad i \in N \\
 & C5: t_{i,j}^{off} > 0 \quad i = 1, 2, \dots, N, j \in M
 \end{aligned} \tag{13}$$

where  $E_{i,j}^{con}(R_{i,j}^{off}, t_{i,j}^{off}) = f(i, j) \sum_{j=1}^M t_{i,j}^{off} + \frac{\kappa_j((R_{i,j} - R_{i,j}^{off})C_j)^3}{T^2} + \frac{t_{i,j}^{off}}{h_{i,j}^m} f(\frac{R_{i,j}^{off}}{t_{i,j}^{off}}) + P_{m,j} t_{i,j}^{off}$ .

Since  $f(x)$  is a convex function with  $x \geq 0$ , its perspective function  $\frac{t_{i,j}^{off}}{h_{i,j}^m} f(\frac{R_{i,j}^{off}}{t_{i,j}^{off}})$  is convex relative to  $R_{i,j}^{off} \geq 0$  and  $t_{i,j}^{off} \geq 0$ . We get that the objective function and constraint 3 of the P4 problem are convex, and the other constraints are convex. Therefore, the P4 problem is convex and can be solved by standard convex optimization techniques.

The detailed solution for solving problem P4 and main problem P1 is described as follows.

### 5.1. Through the Lagrange Method and KKT Conditions to Find the Optimal Solution Properties

Constructing a Lagrangian function usually adds all constraints to the objective function, thereby transforming the constraint problem into an unconstrained problem. However, in this construction process, we only consider most constraints but not all constraints. Let  $u_i = [u_1, u_2, \dots, u_N] \succ 0$ ,  $\lambda_i = [\lambda_1, \lambda_2, \dots, \lambda_N] \succ 0$ ,  $v_{i,j} = [[v_{1,1}, v_{1,2}, \dots, v_{1,M}], [v_{2,1}, v_{2,2}, \dots, v_{2,M}], \dots, [v_{N,1}, v_{N,2}, \dots, v_{N,M}]] \succ 0$  and  $w_{i,j} = [[w_{1,1}, w_{1,2}, \dots, w_{1,M}], [w_{2,1}, w_{2,2}, \dots, w_{2,M}], \dots, [w_{N,1}, w_{N,2}, \dots, w_{N,M}]] \succ 0$  represents the corresponding Lagrange multiplier. The partial Lagrange function of question P4 is defined as:

$$\begin{aligned}
 L(R, t, u, v, w, \lambda) = & \sum_{i=1}^N \sum_{j=1}^M f(i, j) \sum_{i=1}^M t_{i,j}^{off} + \frac{\kappa_j((R_{i,j} - R_{i,j}^{off})C_j)^3}{T^2} + \frac{t_{i,j}^{off}}{h_{i,j}^m} f(\frac{R_{i,j}^{off}}{t_{i,j}^{off}}) + P_{m,j} t_{i,j}^{off} \\
 & + \sum_{i=1}^N u_i (\sum_{j=1}^M R_{i,j} C_j - F) + \sum_{i=1}^N \sum_{j=1}^M v_{i,j} (R_{i,j} - R_{i,j}^{off}) + \sum_{i=1}^N \sum_{j=1}^M w_{i,j} (\frac{\kappa_j((R_{i,j} - R_{i,j}^{off})C_j)^3}{T^2} \\
 & + \frac{t_{i,j}^{off}}{h_{i,j}^m} f(\frac{R_{i,j}^{off}}{t_{i,j}^{off}}) + P_{m,j} t_{i,j}^{off} - (f(i, j)T - f(i, j) \sum_{j=1}^M t_{i,j}^{off})) + \sum_{i=1}^N \lambda_i (\sum_{j=1}^M t_{i,j}^{off} - T)
 \end{aligned} \tag{14}$$

We supposed  $(R^*, t^*)$  to represent the optimal solution of the P4 problem, and  $(u^*, v^*, w^*, \lambda^*)$  represents the optimal Lagrange multipliers. By applying the Karush–Kuhn–Tucker (KKT) condition, we get a relation about the optimal solution:

$$\frac{\partial L(R, t, u, v, w, \lambda)}{\partial t} = (1 + w_{i,j}^*)(f(i, j) + \frac{1}{h_{i,j}^m} f(\frac{R_{i,j}^*}{t_{i,j}^*}) - \frac{1}{h_{i,j}^m} f'(\frac{R_{i,j}^*}{t_{i,j}^*}) \frac{R_{i,j}^*}{t_{i,j}^*} + p_{m,j}) + \lambda_i^* = 0 \tag{15}$$

where  $w_{i,j}^* \geq 0$  and  $\lambda_i^* = 0$  we can get an equation:

$$f(i, j) + \frac{1}{h_{i,j}^m} f\left(\frac{R_{i,j}^*}{t_{i,j}^*}\right) - \frac{1}{h_{i,j}^m} f'\left(\frac{R_{i,j}^*}{t_{i,j}^*}\right) \frac{R_{i,j}^*}{t_{i,j}^*} + p_{m,j} = 0 \tag{16}$$

Let  $h(x) = \frac{1}{a}f(x) - \frac{1}{a}f'(x)x$ , where  $f'(x)$  is the first-order derivative of  $f(x)$ .  $h(x)$  is a monotonic decreasing function of  $x \geq 0$  with  $h(0) = 0$ . Given  $G < 0$ , there exists a unique positive solution for equation  $h(x) = G$ , given by  $x^* = \frac{B}{\ln 2} [W_0(-\frac{aG}{N_0 d_{i,j} e} - \frac{1}{e}) + 1]$ .  $W_0(x)$  is the main branch of the Lambert  $W$  function, which defines  $W_0(x)e^{W_0(x)} = x$ , and  $e$  is the base of the natural logarithm [32]. Bring  $a = h_{i,j}^m$  and  $x^* = \frac{R_{i,j}^*}{t_{i,j}^*}$ , we get  $\frac{R_{i,j}^*}{t_{i,j}^*} = \frac{B}{\ln 2} [W_0(-\frac{aG}{N_0 d_{i,j} e} - \frac{1}{e}) + 1]$ . Therefore, we have:

$$t_{i,j}^* = \frac{R_{i,j}^* \ln 2}{B [W_0(\frac{h_{i,j}^m(f(i,j) + p_{m,j})}{N_0 d_{i,j} e} - \frac{1}{e}) + 1]} \tag{17}$$

### 5.2. Use the Optimal Solution Properties to Rearrange the Problem

Since  $P4$  is proved to be a convex optimization problem, the property of the optimal solution is obtained by the KKT condition. Therefore, the optimal solution of  $P4$  will surely possess the property. By leveraging characteristic (17), the objective function of  $P4$  can be transformed into a polynomial form with only one unknown vector and reducing the constraints accordingly. Put the obtained  $t_{i,j}^*$  into the  $P4$  problem to get the problem  $P5$ :

$$\begin{aligned} P5 \quad & \min_{R^{off}} \sum_{i=1}^N \sum_{j=1}^M E_{i,j}^{con}(R_{i,j}^{off}) \\ \text{s.t.} \quad & C1 : \sum_{j=1}^M R_{i,j}^{off} C_j \leq F \quad i \in N \\ & C2 : (E_{i,j}^{loc} + E_{i,j}^{off}) - E_{i,j} \leq 0 \quad i \in N, j \in M \\ & C6 : 0 < R_{i,j}^{off} \leq \min(R_{i,j}, \frac{TB}{\ln 2} [W_0(\frac{h_{i,j}^m(f(i,j) + p_{m,j})}{N_0 f_{i,j} e} - \frac{1}{e}) + 1]) \quad i \in N, j \in M \end{aligned} \tag{18}$$

We bring  $t_{i,j}^*$  into constraint 4 to get the constraint about  $R_{i,j}^{off}$ , and then combine constraint 2 and constraint 5 to get constraint 6.

In problem  $P5$ , the objective function is in polynomial form and there are three constraints. Therefore,  $P5$  is a nonlinear constrained optimization problem that can be solved by applying several algorithms.

### 5.3. Solving the Simplified Problem through the ALM Method

The augmented Lagrangian method is used to reconstruct the constrained problem into an unconstrained problem, and then solve the unconstrained optimization problem. Augmented Lagrangian function:

$$\begin{aligned} L_C(R, \alpha, \beta, \theta, \mu) = & \sum_{i=1}^N \sum_{j=1}^M E_{i,j}^{con}(R_{i,j}^{off}) + \sum_{i=1}^N \alpha_i (\sum_{j=1}^M R_{i,j}^{off} C_j - F) + \sum_{i=1}^N \sum_{j=1}^M \beta_{i,j} A \\ & + \sum_{i=1}^N \sum_{j=1}^M \theta_{i,j} (-R_{i,j}^{off}) + \sum_{i=1}^N \sum_{j=1}^M \mu_{i,j} D + \frac{c}{2} (\sum_{i=1}^N \|\sum_{j=1}^M R_{i,j}^{off} C_j - F\|_2^2 + \sum_{i=1}^N \sum_{j=1}^M \|A\|_2^2) \\ & + \sum_{i=1}^N \sum_{j=1}^M \|R_{i,j}^{off}\|_2^2 + \sum_{i=1}^N \sum_{j=1}^M \|D\|_2^2 \end{aligned} \tag{19}$$

In addition,  $c$  is the penalty factor. In order to make the formula concise, we make the following expression:

$$A = \frac{\kappa_j((R_{i,j} - R_{i,j}^{off})C_j)^3}{T^2} + \frac{R_{i,j}^{off} \ln 2}{h_{i,j}^m B[W_0(\frac{h_{i,j}^m(f(i,j)+p_{m,j})}{N_0 d_{i,j} e} - \frac{1}{e}) + 1]}$$

$$f(\frac{B[W_0(\frac{h_{i,j}^m(f(i,j)+p_{m,j})}{N_0 d_{i,j} e} - \frac{1}{e}) + 1]}{\ln 2}) + \frac{p_{m,j} R_{i,j}^{off} \ln 2}{B[W_0(\frac{h_{i,j}^m(f(i,j)+p_{m,j})}{N_0 d_{i,j} e} - \frac{1}{e}) + 1]} \tag{20}$$

$$- (f(i, j)T - f(i, j) \sum_{j=1}^M \frac{R_{i,j}^{off} \ln 2}{B[W_0(\frac{h_{i,j}^m(f(i,j)+p_{m,j})}{N_0 d_{i,j} e} - \frac{1}{e}) + 1]})$$

$$D = R_{i,j}^{off} - \min(R_{i,j}, \frac{TB}{\ln 2} [W_0(\frac{h_{i,j}^m(f(i, j) + p_{m,j})}{N_0 f_{i,j} e} - \frac{1}{e}) + 1]) \tag{21}$$

We select the ALM to solve the unconstrained problem (18) in each iteration, the optimal solution is denoted by  $R_{i,j}^{(k^*)}$ .  $k$  is the number of iterations, given the termination limits  $\varepsilon$ , the initial candidate point  $R_{i,j}^{(0)}$ , the initial penalty factor  $c_0$  and the penalty factor growth coefficient  $\rho$ .

The penalty factor will be updated iteratively until the termination condition  $c_k > \varepsilon$  or the iteration time  $k$  is larger than 50, and then the optimal solution  $R_{i,j}^{(k^*)}$  is obtain. The pseudo code of the ALM algorithm in Algorithm 1:

---

**Algorithm 1** Augmented Lagrange Method

---

**Input:**  $R_{i,j}^{(0)}$  : initial candidate point;  $c_k$  : penalty factor;  $\varepsilon$  : termination limits;  $\rho$  : growth coefficient of penalty factor;  $k = 0$  : iteration times

- 1: **repeat**
- 2: solving the unconstrained problem (18) by ALM method, get  $R_{i,j}^{(k^*)}$ ;
- 3:  $c_{k+1} = \rho c_k$ ;
- 4:  $k = k + 1$ ;
- 5: **until**  $c_k > \varepsilon$  or  $k = 50$

**Output:** optimal  $R_{i,j}^{(k^*)}$

---

5.4. Obtaining Other Optimization Variables

Based on the optimal offloading task size obtained by the algorithm, the optimal offloading time can be obtained by (17), the optimal harvest time can be obtained by  $t_i^{har} = T - \sum_{j=1}^M t_{i,j}^{off}$  and the CPU frequency can also be obtained by  $f_{i,j}^* = \frac{(R_{i,j} - R_{i,j}^{off})C_j}{T}$ .

6. Simulation Results

In this section, we conduct simulation experiments to verify the advantages of the proposed joint optimization scheme, which aims at residual energy maximization of multi MDs in multi-time slots, and jointly optimizes the amount of task-offloading, energy harvesting time, task-offloading time and CPU frequency. All simulation programs are written in the Python programming language and use the PyOpt, which is a Python-based software package for solving optimization problems with nonlinear constraints.

The numerical results are compared with the following two benchmark schemes to evaluate the joint optimization scheme for maximizing residual energy under multi-MDs and multi-slots. The scheme jointly optimizes the task-offloading size, energy harvesting time, task-offloading time and the CPU frequency, which is abbreviated as MMM.

1. Local computation only: Each user completes its computation task only through local computations, and the entire time slice is used to harvest energy for each user, referred to as LOC.
2. Only computation offload: Each user completes his computing task only by completely offloading the computing task to the AP, and the energy harvested by the user is used to support computing offload, referred to as OFF.
3. Fixed energy harvesting time: The energy harvesting time of each user is fixed [28], and the task-offloading size and task-offloading time are optimized, referred to as FEH.
4. Fixed task offloading size: The size of each user’s task-offloading is fixed [28], which can optimize energy harvesting time and offloading time, referred to as FTO.

In the simulation, we first used to set the general parameters of the simulation. The system parameters are setting as follows (unless otherwise stated): The number of the MDs  $M = 3$ , the number of time slots  $N = 3$ , the period required for 1-bit calculation is subject to uniform distribution  $C_j \in [600, 800]$  cycles/bit,  $\kappa_j = 10^{-28}$ ,  $\forall j \in M$  [13], the maximum CPU frequency  $f_j^{max} = 1$  GHz, the AP transmission power  $p_{i,j}^a = 200$  W, the circuit power  $p_{m,j} = 10^{-4}$  W, the receiver noise power  $N_0 = 10^{-9}$  W and the spectrum bandwidth for offloading  $B = 2$  MHz. The distance between the MD and the AP follow  $d_{i,j} \sim U(5.0, 8.0)$  m, the task size follow  $R_{i,j} \sim U(50, 80)$  kb and the device received energy conversion rate follows the equal division distributed  $v \in [0.7, 0.9]$ . The channel power gain from each MD to the AP is modeled as  $h_{i,j}^m = 10^{-3} d_{i,j}^{-\alpha} \phi_i$ , where  $\phi_i$  represents the short-term fading which is assumed to be an exponentially distributed random variable with unit mean,  $\alpha$  denotes the path loss exponent and here set  $\alpha = 2$  and the channel power gain  $h_{i,j}^a$  from the AP to each MD is equal to  $h_{i,j}^m$ . In addition, set the time block  $T = 2$  s .

### 6.1. Performance of Solution

The first simulation shows the relationship between the residual energy (J) and the AP transmission power (W). By observing the changes of the residual energy with the AP transmission power during the five methods, it is assumed that  $d_{i,j}, R_{i,j}, v_j$  are transmitted randomly according to the above. The results are shown in Figure 3.

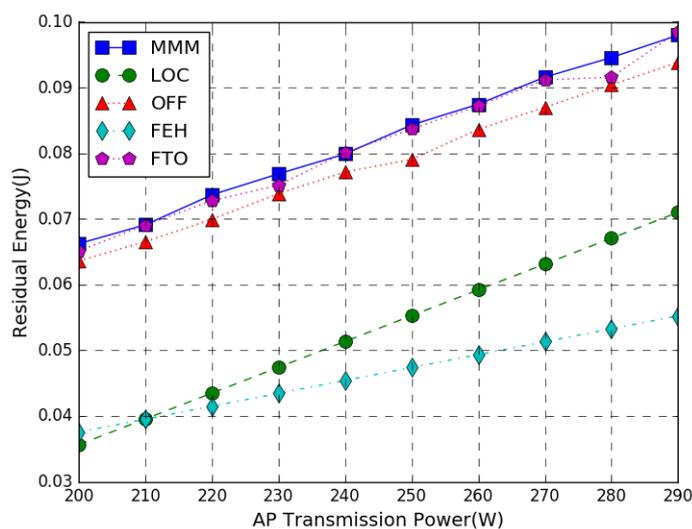


Figure 3. Access point (AP) transmission power vs. residual energy.

From Figure 3, we can observe that when the AP transmission power varies from 200 W to 290 W, the MMM and FTO curves always have relatively large residual energy. However, the MMM curve shows better stability because the proposed joint optimization scheme considers the task-offloading size, while FTO fixes the task-offloading size and only optimizes the task-offloading time and energy harvesting time. In contrast, OFF offloads all tasks to the AP, which means that the size of task offload is fixed, and optimize the only task-offloading time and energy harvesting time, which requires reducing energy harvesting time and increasing task-offloading time. Similarly, although LOC does not require task offload size and task offload time, it only needs to use time slots for harvesting. We all know that MDs do not have strong computing power. Of course, the residual energy is more than FEH only. The FEH solution fixes the energy harvesting time, which does not take so long for task-offloading. Therefore, the energy harvesting time is shortened, the energy consumption increased further the smallest residual energy. In addition, we found that MMM and FTO intersect among the five curves, OFF is closer to MMM, and the LOC and FEH curves are a straight line.

The second simulation shows the relationship between the residual energy (J) and the task size (KB). The tasks we declared earlier are randomly selected. For the sake of the intuitiveness of the task size changes, we directly give a fixed task size change. There are multiple MDs in a time block, assuming that each MD has the same task size. By changing the size of the task, we observed the changes in the residual energy when the five methods were executed separately. The results are shown in Figure 4.

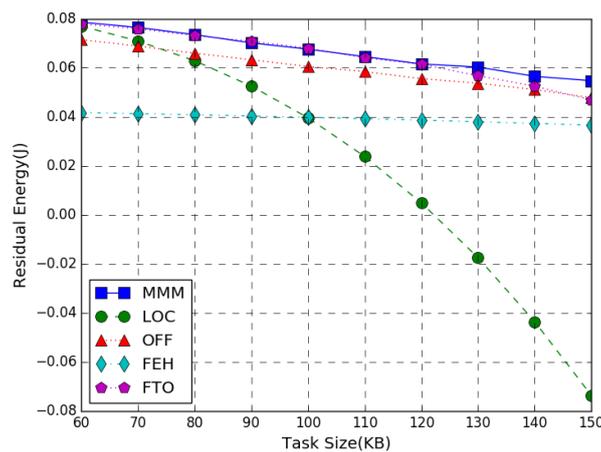


Figure 4. Task size vs. residual energy.

It can be seen from the results in Figure 4 that no matter which optimization scheme is adopted, the residual energy decreases with the increase of the task size, but the proposed joint optimization scheme is still superior to other benchmark schemes and has better stability than the benchmark scheme. Specifically, when the task size is bigger, the decline speed of the LOC curve is faster, while the decline speed of MMM, OFF and FTO is very slow. However, the residual energy of MMM is always greater than OFF, and the FTO curve decreases faster when the task is large, which shows that MMM has a good solution combination and better ability to adapt to larger tasks. MMM jointly optimizes task-offloading size, energy harvesting time and task-offloading time to find the best combination, while LOC and OFF can only execute all tasks in the local and MEC servers, respectively. FEH optimizes the task-offloading size, fixes the energy harvesting time and the total offloading time of all equipment tasks. Therefore, the residual energy of FEH is relatively stable as the task increases. In addition, when the task is relatively small, the residual energy of the other four methods except FEH tends to be equal, and the residual energy of FEH is only half of them, which shows that fixing energy harvesting time is unreasonable.

The third simulation shows the relationship between the residual energy (J) and the length of the time block  $T$ (s). By changing the length of the time block, we observe the residual energy changes when the five schemes are executed separately. The results are shown in Figure 5.

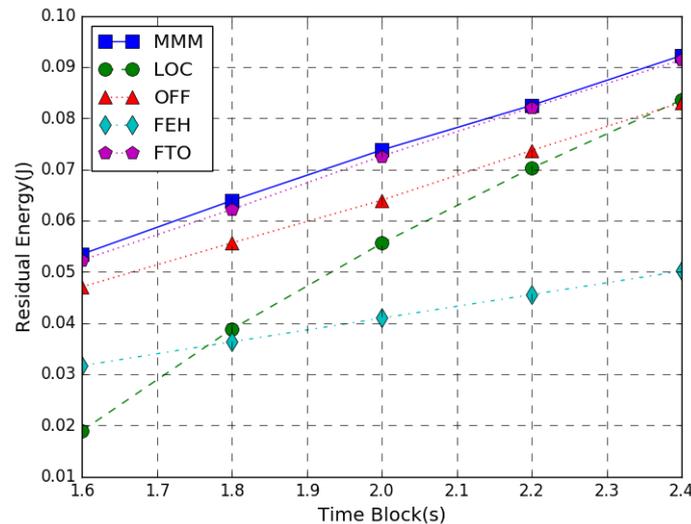


Figure 5. Time block vs. residual energy.

From Figure 5, we have several observations. First, when the time block is small relatively, the residual energy of the LOC is the smallest. The energy harvesting time at this time is relatively short, and the energy consumed by the local execution task is relatively large. At the same time, the residual energy of MMM and FTO are very close to some extent, which indicates that the combination of FTO and MMM solutions is similar. Secondly, as the length of the time block increases, the residual energy of the LOC quickly approaches that of OFF, which indicates that the combination of OFF method solutions is no close to the MMM optimal combination, and the rapid growth of LOC does not quickly approach MMM. Finally, we found that the curves of MMM and OFF and FTO are closer to a straight line, but the linear rise of MMM and FTO is larger, which shows that the better performance and adaptability of MMM and FTO schemes. In addition, as the time block becomes increases, the growth rate of the FET is the smallest. From the results, compared with other solutions, the residual energy is only approximately half compared with other solutions.

### 6.2. Failure Ratio of Solution

The simulation shows the relationship between failure ratio (%) and task size. By changing the size of the task, we studied the failure ratio of the five scenarios. Failure ratio means that the residual energy is less than 0 or the task is not complete within time  $T$ . Assuming the task size follows from 60 KB to 150 KB, as the task size increases, the result of the failure ratio is shown in Figure 6.

It can be seen in Figure 6 that the failure ratio grows with the increase in tasks. First, the failure ratio is very high when the task is very large, but the proposed MMM has a relatively low failure ratio. MMM can find the best combination of task-offloading size, energy harvesting time and task-offloading time. Second, the curves of FEH and OFF are very tortuous. When the task is small, the failure ratio is the highest. When the task is large, the failure ratio is almost equal and relatively high, and it is higher than the failure ratio of MMM only. Then, the growth rate of the failure ratio of the FTO curve increases rapidly as the task increases, especially when the task is relatively large, the failure rate approaches 1. The fixed offload task size makes the number of local executed tasks too large to be completed within  $T$ . MMM can reasonably divide the task offload size, thereby allowing users to complete tasks within time  $T$ . Finally, when the task is small, the failure rate of the LOC can almost reach zero. The local parallel execution meets the demand, but the task offload of the MMM scheme is

serial execution, which will cause relatively more energy consumption. When the task is large enough, the failure rate of LOC is as high as 100%. When the task is small, the harvesting energy can fully satisfy the task to execute locally. As the task increases, the harvesting energy cannot meet the large amount of energy consumed by the local execution.

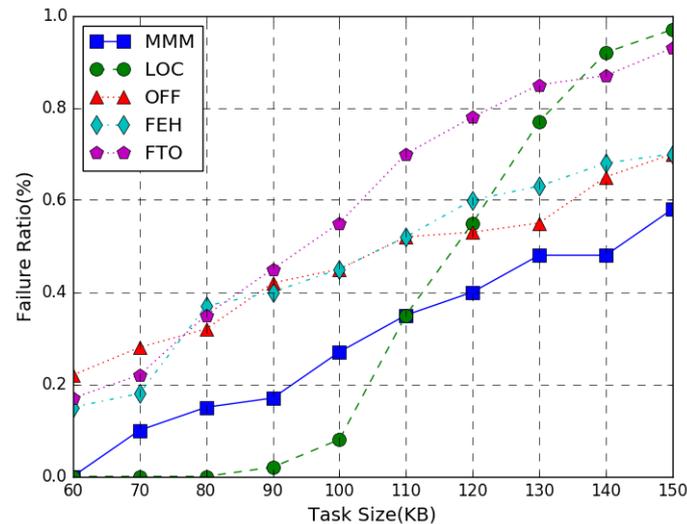


Figure 6. Task size vs. failure ratio.

## 7. Conclusions

In this article, we propose an MEC and WPT integrated system that considers multiple users and partial task-offloading in multiple time slots. The multi-antenna AP provides energy for multiple MDs at the same time, and the energy harvested by the MDs provides energy for task-offloading and local computing. By joint optimization of task-offloading size, energy harvesting time, task-offloading time and CPU frequency, it is ensured that the energy harvested by each MD in each time block meets the energy consumption requirements of various tasks. At the same time, it can extend the battery duration of MDs and improve the computing power of MDs. The optimal solution is obtained by using the Lagrange method, KKT condition and augmented Lagrangian method. The numerical results show that the proposed joint optimization scheme is superior to the traditional benchmark scheme. The integrated system of MEC and WPT is becoming the hotspot research because of its practicability. Our work provides a new idea for related research. Next, the development of MDs will have greater applicability. In the following research, we will pay more attention to the nonlinear energy transmission model, and apply more task offloading communication protocols such as Orthogonal Frequency Division Multiple Access to make the system model more in line with technological development.

**Author Contributions:** Conceptualization, L.L. and Y.L.; methodology, L.L. and G.X.; software, L.L. and Y.L.; validation, L.L. and Y.L.; investigation, L.L. and P.L.; resources, Y.L.; data curation, L.L.; writing—original draft preparation, L.L.; writing—review and editing, L.L., P.L., Y.L. and J.G.; visualization, L.L.; supervision, G.X.; project administration, G.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by National Key R&D Plan of China under Grant 2017YFA0604500, and in part by the Jilin Province Science and Technology Development Plan Project under Grant 20190201180JC, 20200401076GX, in part by the Jilin Province Education Department '13th Five-Year' Science and Technology Project under Grant JJKH20191310KJ.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chiang, M.; Zhang, T. Fog and IoT: An Overview of Research Opportunities. *IEEE Internet Things J.* **2017**, *3*, 854–864. [\[CrossRef\]](#)
2. Ahmed, A.; Ahmed, E. A Survey on Mobile Edge Computing. In Proceedings of the International Conference on Intelligent Systems & Control, Coimbatore, India, 7–8 January 2016; pp. 1–8. [\[CrossRef\]](#)
3. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [\[CrossRef\]](#)
4. Lu, X.; Wang, P.; Niyato, D.; Kim, D.I.; Han, Z. Wireless Networks With RF Energy Harvesting: A Contemporary Survey. *Commun. Surv. Tutor. IEEE* **2015**, *17*, 757–789. [\[CrossRef\]](#)
5. Bi, S.; Zhang, Y.J.A. Computation Rate Maximization for Wireless Powered Mobile-Edge Computing with Binary Computation Offloading. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 4177–4190. [\[CrossRef\]](#)
6. Bi, S.; Ho, C.K.; Zhang, R. Wireless Powered Communication: Opportunities and Challenges. *IEEE Commun. Mag.* **2015**, *53*, 117–125. [\[CrossRef\]](#)
7. Kumar, K.; Lu, Y.H. Cloud Computing for Mobile Users: Can Offloading Computation Save Energy? *Computer* **2010**, *43*, 51–56. [\[CrossRef\]](#)
8. Eduardo, C.; Aruna, B.; Dae-Ki, C.; Alec, W.; Paramvir, B. MAUI: Making smartphones last longer with code offload. In Proceedings of the International Conference on Mobile Systems, San Francisco, CA, USA, 15–18 June 2010; pp. 49–62. [\[CrossRef\]](#)
9. Byung-Gon, C.; Sunghwan, I.; Petros, M.; Mayur, N.; Ashwin, P. CloneCloud: Elastic Execution between Mobile Device and Cloud. In Proceedings of the Conference on Computer Systems ACM, San Jose, CA, USA, 10–13 April 2011; pp. 301–314. [\[CrossRef\]](#)
10. Kosta, S.; Aucinas, A.; Hui, P.; Mortier, R.; Zhang, X. ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In Proceedings of the IEEE Infocom, Orlando, FL, USA, 25–30 March 2012; pp. 945–953. [\[CrossRef\]](#)
11. Pavel, M.; Zdenek, B. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *Commun. Surv. Tutor. IEEE* **2017**, *19*, 1628–1656. [\[CrossRef\]](#)
12. Ren, J.; Yu, G.; Cai, Y.; He, Y. Latency Optimization for Resource Allocation in Mobile-Edge Computation Offloading. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 5506–5519. [\[CrossRef\]](#)
13. You, C.; Huang, K.; Chae, H.; Kim, B.H. Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1397–1411. [\[CrossRef\]](#)
14. You, C.; Zeng, Y.; Zhang, R.; Huang, K. Asynchronous Mobile-Edge Computation Offloading: Energy-Efficient Resource Management. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 7590–7605. [\[CrossRef\]](#)
15. Avgeris, M.; Dechouniotis, D.; Athanasopoulos, N.; Papavassiliou, S. Adaptive Resource Allocation for Computation Offloading: A Control-Theoretic Approach. *ACM Trans. Internet Technol.* **2019**, *19*, 1–20. [\[CrossRef\]](#)
16. Wang, Y.; Sheng, M.; Wang, X.; Wang, L.; Li, J. Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling. *IEEE Trans. Commun.* **2016**, *64*, 4268–4282. [\[CrossRef\]](#)
17. Al-Shuwaili, A.; Simeone, O. Energy-Efficient Resource Allocation for Mobile Edge Computing-Based Augmented Reality Applications. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 398–401. [\[CrossRef\]](#)
18. Barbarossa, S.; Sardellitti, S.; Lorenzo, P.D. Joint allocation of computation and communication resources in multiuser mobile cloud computing. In Proceedings of the 2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Darmstadt, Germany, 16–19 June 2013; pp. 26–30. [\[CrossRef\]](#)
19. Wang, C.; Liang, C.; Yu, F.R.; Chen, Q.; Tang, L. Computation Offloading and Resource Allocation in Wireless Cellular Networks With Mobile Edge Computing. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 4924–4938. [\[CrossRef\]](#)
20. Xing, H.; Liu, L.; Xu, J.; Nallanathan, A. Joint Task Assignment and Wireless Resource Allocation for Cooperative Mobile-Edge Computing. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6. [\[CrossRef\]](#)
21. Wang, P.; Yao, C.; Zheng, Z.; Sun, G.; Song, L. Joint Task Assignment, Transmission, and Computing Resource Allocation in Multilayer Mobile Edge Computing Systems. *IEEE Internet Things J.* **2019**, *6*, 2872–2884. [\[CrossRef\]](#)

22. Wang, F.; Xu, J.; Wang, X.; Cui, S. Joint Offloading and Computing Optimization in Wireless Powered Mobile-Edge Computing Systems. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 1784–1797. [[CrossRef](#)]
23. Wang, F.; Xing, H.; Xu, J. Optimal Resource Allocation for Wireless Powered Mobile Edge Computing with Dynamic Task Arrivals. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–7. [[CrossRef](#)]
24. You, C.; Huang, K.; Chae, H. Energy Efficient Mobile Cloud Computing Powered by Wireless Energy Transfer. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 1757–1771. [[CrossRef](#)]
25. Mao, Y.; Zhang, J.; Letaief, K.B. Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3590–3605. [[CrossRef](#)]
26. Gao, H.; Ejaz, W.; Jo, M. Cooperative Wireless Energy Harvesting and Spectrum Sharing in 5G Networks. *IEEE Access* **2016**, *4*, 3647–3658. [[CrossRef](#)]
27. Hu, X.; Wong, K.; Yang, K. Wireless Powered Cooperation-Assisted Mobile Edge Computing. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 2375–2388. [[CrossRef](#)]
28. Liu, P.; Xu, G.; Yang, K.; Wang, K.; Li, Y. Joint Optimization for Residual Energy Maximization in Wireless Powered Mobile-Edge Computing Systems. *KSII Trans. Internet Inf. Syst.* **2018**, *12*, 5614–5633. [[CrossRef](#)]
29. Tao, X.; Ota, K.; Dong, M.; Qi, H.; Li, K. Performance Guaranteed Computation Offloading for Mobile-Edge Cloud Computing. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 774–777. [[CrossRef](#)]
30. Zhou, F.; Hu, R.Q. Computation Efficiency Maximization in Wireless-Powered Mobile Edge Computing Networks. *IEEE Trans. Wirel. Commun.* **2020**, *99*. [[CrossRef](#)]
31. Boyd, S.; Boyd, S.P.; Vandenberghe, L. Convex Optimization. *IEEE Trans. Autom. Control* **2006**, *51*, 1859. [[CrossRef](#)]
32. Corless, R.M.; Gonnet, G.H.; Hare, D.E.G.; Jeffrey, D.J.; Knuth, D.E. On the Lambert W Function. *Adv. Comput. Math.* **1996**, *5*, 329–359. [[CrossRef](#)]

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).