Joint Optimization for Residual Energy Maximization in Wireless Powered Mobile-Edge Computing Systems

Peng Liu[1], Gaochao Xu[1], Kun Yang[2], Kezhi Wang[3], Yang Li[1,*]
[1] College of Computer Science and Technology, Jilin University
Changchun, Jilin 130012, P.R. China
[e-mail: pliu16@mails.jlu.edu.cn]
[2] School of Computer Science and Electronic Engineering, University of Essex
Wivenhoe Park, Colchester, CO4 3SQ, UK
[e-mail: kunyang@essex.ac.uk]
[3] Department of Computer and Information Sciences Northumbria University
Newcastle upon Tyne, NE1 8ST, UK
[e-mail: kezhi.wang@northumbria.ac.uk]
*Corresponding author: Yang Li

Received December 29, 2017; revised April 30, 2018; accepted July 12, 2018; published December 31, 2018

Abstract

Mobile Edge Computing (MEC) and Wireless Power Transfer (WPT) are both recognized as promising techniques, one is for solving the resource insufficient of mobile devices and the other is for powering the mobile device. Naturally, by integrating the two techniques, task will be capable of being executed by the harvested energy which makes it possible that less intrinsic energy consumption for task execution. However, this innovative integration is facing several challenges inevitably. In this paper, we aim at prolonging the battery life of mobile device for which we need to maximize the harvested energy and minimize the consumed energy simultaneously, which is formulated as residual energy maximization (REM) problem where the offloading ratio, energy harvesting time, CPU frequency and transmission power of mobile device are all considered as key factors. To this end, we jointly optimize the offloading ratio, energy harvesting time, CPU frequency and transmission power of mobile device to solve the REM problem. Furthermore, we propose an efficient convex optimization and sequential unconstrained minimization technique based combining method to solve the formulated multi-constrained nonlinear optimization problem. The result shows that our joint optimization outperforms the single optimization on REM problem. Besides, the proposed algorithm is more efficiency.

Keywords: mobile edge computing, wireless power transfer, joint optimization, convex optimization, BFGS

http://doi.org/10.3837/tiis.2018.12.001 **ISSN : 1976-7277**

1. Introduction

With the explosive development of the mobile device and mobile communication, mobile devices are becoming more and more widely used and the novel mobile applications emerge continuously. However, the following is the great demand of physical resource of mobile devices, which is just the key bottleneck of current mobile device. To tackle the problem, researchers proposed the framework of Mobile Cloud Computing (MCC)[1,2], it migrates the computation-intensive task to the cloud, leveraging the sufficient resources of data centers to finish the computation. Nevertheless, the user experience under the MCC framework largely depends on the network, often bringing the high latency. In order to increase the bandwidth and decrease the latency, Mobile Edge Computing (MEC) is proposed[3,4]. In the MEC, computation resources are provided within the Radio Access Network (RAN) in close proximity to the mobile device. Due to the short distance, the MEC paradigm can achieve low latency, high bandwidth and computing agility[5].

Unfortunately, although the MCC and MEC bring the powerful computation resources for the mobile devices, it still faces the problem of insufficient battery energy. Prolonging the battery lives is a big challenge since the advent of smart mobile devices, but still has no breakthrough. Recently, the development of radio frequency (RF) based wireless power transfer (WPT) technology bring a new solution to solve the problem of insufficient battery energy[6]. WPT uses dedicated RF energy transmitter, which can continuously power the battery of remote energy-harvesting (EH) devices[7].

In this paper, we integrate EH into MEC, where the mobile device is able to harvest energy and execute the computation-intensive task on the edge server and local device parallelly during each time block. In the proposed integrated system, we focus on jointly optimizing the communication and computation resources for the partial computation offloading system. Considering the special situation of this paper, we not only jointly optimize the transmission power for communication and the CPU frequency for computation, but also optimize the offloading ratio and the energy harvesting time for each mobile device. Unlike previous researches that pay more attention to the interest of operators, this paper mainly focus on the user interest, especially for prolonging battery life of mobile device. For this purpose, by the proposed jointly optimizing scheme, we aim at solving the residual energy maximization (REM) and the energy consumption minimization (ECM) problems at the same time. Finally, we formulate the two problems as a nonlinear programming problem with inequality constraints and design a convex optimization and sequential unconstrained minimization technique based combining method to solve it. The simulation results show that the proposed jointly optimizing scheme outperforms the previous ones in the partial computation offloading system, further, proposed algorithm is proven to be more efficiency.

This paper is organized as follows: The related work is introduced in section II. In section III, we introduce the WPT and MEC integrated system model. Section IV presents the problem formulation and analysis. Section V gives the algorithm for solving the formulated problem. Sections VI shows the simulation results and Section IV gives the conclusion.

2. Related Work

As the fundamental policy of MEC and MCC, computation offloading has attracted more and more attention in recent years. Computation offloading is able to provide enough physical resources and save energy for the mobile devices. [8] verified that offloading operation to the cloud can potentially save energy and extend battery lifetimes for mobile users, and [9,10,11] proposed different computation offloading frameworks. Besides, there are two basic computation task offloading models in MEC, i.e. binary and partial computation offloading, where binary offloading requires a task to be executed as a whole either locally at the mobile device or remotely at the edge server and partial offloading allows a task to be partitioned into two parts with one executed locally and the other offloaded for edge execution[12].

However, the efficiency of computation offloading largely depends on the quality of communication because data transmission is necessary. This calls for incorporating the characteristics of communication and computation, jointly optimizing communication and computation resources[13,14,15,16,17]. In [13], a method jointly optimizing mobile-transmission power and CPU cycles assigned to each application is proposed to minimize the power consumption at the mobile side, under an average latency constraint. By KKT condition, they get the one-to-one relationship between the transmit power and the percentage of CPU cycles assigned to each user. In [14], the author considered a single-user scenario, they focus on partial computation offloading by jointly optimizing the CPU cycle frequency, transmission power and offloading ratio. They proposed two algorithms to handle the energy consumption minimization problem and latency minimization problem respectively. In [15], the radio resource and computational resource allocation were jointly optimized to minimize the weighted sum energy consumption in a MIMO system and then proposed a successive convex approximation based iterative algorithm to solve the problem. In [16], the author jointly optimized the offloading selection, radio resource allocation and computational resource allocation coordinately to minimize the energy consumption of mobile device. They following proposed two methods to solve the formulated Mixed Integer Nonlinear Programming problem. [17] developed an online joint radio and computational resource management algorithm for multi-user MEC systems, they aimed at minimizing the long-term average weighted sum power consumption of the mobile device and MEC server, under the task buffer stability constraint.

By jointly optimizing the computation and communication resources allocation, the above researches outperform the conventional single optimization method. However, all the researches only focus on saving energy, but ignoring the new technology WPT which can power the mobile device. Some researches have started to integrate energy harvest into offloading process[18,19,20,21,22]. Powering mobile devices in MEC systems with wireless energy harvesting was earlier proposed in [18]. They considered the wireless powered single-user MEC system with binary offloading, where the authors aimed at maximizing the probability of successful computation, by deciding whether a task should be fully offloaded or not, subject to the computation latency constraint. In [19], the author considered a more general case with more than one user, and allowing for more flexible partial offloading to improve the system performance in terms of the energy efficiency. Specifically, authors in this paper developed an optimal resource allocation scheme that jointly optimizing the energy transmit beamformer, the CPU frequency, the numbers of offloaded bits as well as the time allocation among users to minimize the AP's total energy consumption under the user's individual computation latency constraints. In [20], the author also investigated MEC system with EH mobile device, but they aimed at minimizing the execution cost, which consists of execution delay and task failure. To solve the problem, they proposed a low-complexity online algorithm based on Lyapunov optimization. However, these two works only considered the smaller scale system with one mobile device. [21] considered a multi-user MEC network powered by WPT, where the wireless devices follow binary offloading policy, aiming at maximizing the sum computation rate of all the wireless devices in the network. To this end, they proposed a decoupled optimization method and an ADMM based decomposition technique to solve the problem. [22] proposed a novel best cooperative mechanism (BCM) for wireless energy harvesting and spectrum sharing in 5G networks. They follwing formulated an optimization problem based on BCM with the objective to maximize the throughput of users.

There are not many related works on MEC integrated EH system currently, and these works achieve different purpose by considering diverse system model, optimization parameters and methods. To our best knowledge, there is no work with the goal of maximizing the residual energy of mobile devices in such MEC and EH integrated system, which is able to guarantee the higher energy harvest efficiency and lower energy consumption at the same time. Accordingly, this paper with the goal of maximizing the residual energy of system by jointly optimizing the parameters during the process of computation, computation and energy harvest.

3. System Model

This paper considers a MEC and EH integrated system with multi-user, supposing that there are one AP with multi-antenna and N mobile devices with single-antenna. AP is directly connected to edge servers, it can not only receive the task data and transmit them to edge servers, but also be able to power the mobile device by the WPT. The mobile device is not only able to offloading and execute tasks as common ones, but also to harvest energy coming from AP. However, the mobile device can't harvest energy and offloading simultaneously because of the single-antenna [18]. The system architecture is as Fig. 1.

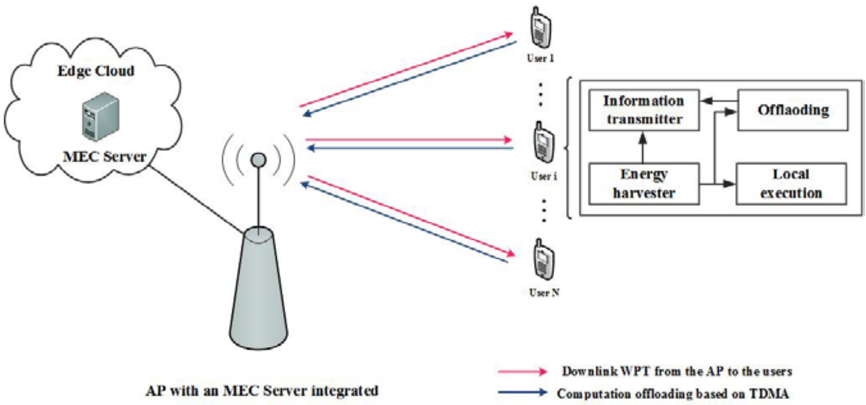


Fig. 1. System architecture

The integrated system is based on a block-based time-division-multiple-access(TDMA) model, the length of each time block is T in which the channel remains static, but varies among different time block. For each task, it can be divided into two parts, one for local execution and the other for offloading. For offloading part, the users firstly harvest the energy at beginning of each time-block and then offloading the task to the edge server within the left time. However, for the local execution part, the execution can be started at the beginning and is able to use up the whole time block. Supposing that the time requirement for task execution is equal to T. The time division for different execution model is as Fig. 2.

Assuming that the AP has perfect knowledge of the channels from AP to users and the computation requirement for each task, and thus be able to coordinate the task partition, computation offloading and local computing by jointly allocating both communication and computation resources.

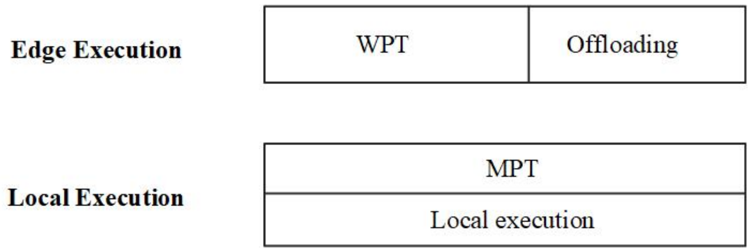


Fig. 2. Time division for different execution model

3.1 Task Model

Supposing that each mobile device has a computation-intensive task U_i , denoted by the following:

$$U_i = (I_i, X_i, T_i) \quad i = 1, 2, 3, ..., N \tag{1}$$

where T_i and I_i denotes the task time constraints and computation input bits of the task U_i, respectively. Supposing that the time constraints for each task are identical and equals to the length of time block, that is, $T = T_i, i = 1, 2, 3, ..., N$. X_i denotes the cycles required to process one bit input, which is depending on the task characteristic, e.g. computation complexity and can be obtained through off-line measurement [23]. By the definition above, the number of cycles required for the U_i is denoted by $I_i X_i$. Besides, allowing for the tasks are partition allowable, executing on the mobile device and edge server simultaneously, we define $\lambda_i (0 \le \lambda_i \le 1)$ as the ratio of edge execution amount of bits to the total input data bits for the i-th task and surely 1-λ_i as the ratio of local execution. To simplify the analysis, we assume that the task can be partitioned into two parts of any size [24], despite that only several partitions are reasonable in practice because of the interdependence. Accordingly, the optimal solution of this paper should be preprocessed before using in practice environment. The processing method is also given in this paper. In addition, the task model ignores the denotation of output data, because they are usually much smaller compared with the input data.

3.2 Local Execution Model

From the definition of task model, the i-th task need $(1-\lambda_i) I_i X_i$ cycles for local execution in total. Supposing that the CPU frequency of i-th mobile device is denoted by $f_{m,i}$, which can be adjusted by the DVFS [25] technique to satisfy the time constraints at the expense of energy consumption growing [26]. However, the frequency can't exceed the limit of the maximum CPU frequency, denoted by $f_{m,i}^{\max}$, which is dependent on chip architecture. The constraint is given by:

$$0 < f_{m,i} \le f_{m,i}^{\max} \quad i = 1, 2, 3, ..., N \tag{2}$$

With the frequency $f_{m,i}$, the local computing time of task U_i are as follows:

$$T_i^{loc} = \frac{(1-\lambda_i) I_i X_i}{f_{m,i}} \quad i = 1, 2, 3, ...N \tag{3}$$

Because the local computing part of each task has to be finished within T, and hence the constraint of execution time is given by:

$$T_i^{loc} \le T \ \ i = 1, 2, 3, ...N \tag{4}$$

We model that the energy consumption per cycle is $\kappa^c f_{m,i}^2$, where κ^c is the effective switched capacitance that depends on the chip architecture, and thus the total energy consumption for local execution part of U_i are as follows:

$$E_i^{loc} \le \kappa^c f_{m,i}^2 (1\text{-}\lambda_i) I_i X_i \ \ i = 1, 2, 3, ...N \tag{5}$$

where we set κ^c=$10e^{-28}$ according to the practical measurement.

3.3 Mobile-Edge Execution Model

The partition size for mobile-edge execution is $\lambda_i I_i$ bits for the i-th task, which should be transmitted to the edge server. Assuming that a high-speed multi core CPU is available at the edge server, hence execution time on the edge server is ignored. We further assume the transmission time for feedback is negligible because of the smaller size of computation output. Accordingly, the energy and time consumption on edge server execution is dependent on communication, in other words, the process that offloading data to edge server. The achievable rate for offloading depends on transmission power and channel state information which will retain static within one time block. According to the Shannon-hartely formula, the achievable rate for the i-th mobile device is as follows:

$$S_i = B\log(1 + \frac{p_{m,i} h_{m,i}}{\sigma^2 d_i}) \ \ i = 1, 2, 3, ...N \tag{6}$$

where B and σ^2 denote the system bandwidth and noise power at the receiver, respectively. $p_{m,i}$ denotes the transmit power of i-th mobile device, $h_{m,i}$ is the channel power gain from mobile device i to the AP, d_i is the distance between mobile device i and AP. Consequently, the time consumption on mobile-edge execution that equals the transmission time for the input and can be given by:

$$T_i^{edge} = \frac{\lambda_i I_i}{S_i} \ \ i = 1, 2, 3, ..., N \tag{7}$$

and also the energy consumption of mobile device i-th for mobile-edge execution is all for offloading, which can accordingly be given by:

$$E_i^{edge} = p_{m,i} \frac{\lambda_i I_i}{S_i} \ \ i = 1, 2, 3, ..., N \tag{8}$$

The computation ability of edge servers can't be unlimited, and hence we define F as the maximal computation ability of the edge server, which in this paper means the maximal number of available CPU cycles in one time block. Then, the total cycles executed on the edge server is constrained by:

$$\sum_{i=1}^{N} \lambda_i I_i X_i \le F \tag{9}$$

3.4 Energy Harvest Model

The multi-antenna AP uses beamforming to transfer power to the mobile device at the beginning of each time block by WPT. Allowing for the technical limitations that charge efficiency is quite low, we assume that the mobile devices have enough battery capacity to store harvested energy, and thus the energy harvested by *i-th* mobile device during the harvest period is as follows:

$$E_i^{charge} = v p_a h_{a,i} t_i \quad i = 1, 2, 3, ..., N \tag{10}$$

where $h_{a,i}$ is the scalar channel power gain of the downlink channel from AP to i-th mobile device, p_a denotes the AP transmission power, and constant $0 < v < 1$ represents the energy conversion efficiency. During each time block, the mobile device firstly harvests energy and then offloads the task within rest time of this block to satisfy the time constraints. By t_i denotes the harvest time and thus maximum available time for offloading is $T - t_i$. Then, the energy harvesting time and offloading time have to satisfy the following constraint:

$$T_i^{edge} + t_i \le T \quad i = 1, 2, 3, ..., N \tag{11}$$

It is assumed that the mobile device masters the accurately estimated of p_a and $h_{a,i}$ which are used for coordinate the offloading execution and local execution. Besides, the conversion efficiency of each device is assumed to be same.

4. Problem Formulation

The goal of this paper is to maximize the residual energy of all the mobile devices within the AP control area, which requires to maximize the harvested energy and minimize the consumed energy at the same time. The amount of harvested energy is directly proportional to the harvest time t_i, larger t_i will surely harvest more energy but will 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, we should coordinate the offloading ratio of input bits that executed on the edge server, the CPU frequency for local execution and transmission power for edge execution to minimize the energy consumption. Accordingly, in order to maximize the residual energy, we need to jointly optimize the harvest time, offloading ratio, CPU frequency and transmission power to find an optimal combination. In this section, we give the formulation of the described residual energy maximization (REM) problem.

4.1 Residual Energy Maximization Problem

Based on the given model, the residual energy of each mobile device denoted by the function $E_i^{res}(f_{m,i}, p_{m,i}, \lambda_i, t_i)$ can be given by:

$$E_i^{res}(f_{m,i}, p_{m,i}, \lambda_i, t_i) = E_i^{charge} - (E_i^{loc} + E_i^{edge}) \tag{12}$$

where the E_i^{charge}, E_i^{loc}, E_i^{edge} is given by (5),(8),(10) respectively. And the detail function expression with the optimization variable can be given by:

$$E_i^{res}(f_{m,i}, p_{m,i}, \lambda_i, t_i) = vp_a h_{a,i} t_i - (p_{m,i} \frac{\lambda_i I_i}{B\log(1+\frac{p_{m,i}h_{m,i}}{\sigma^2 d_i})} + \kappa^c f_{m,i}^2 (1-\lambda_i) I_i X_i) \tag{13}$$

Thus, the REM problem can be formulated as follows:

$$\begin{aligned}
&\text{P1} \quad \max_{\lambda,p,f,t} \sum_{i=1}^{N} E_i^{res}(f_{m,i}, p_{m,i}, \lambda_i, t_i) \\
&\text{s.t.} \quad C1: \frac{\lambda_i I_i}{B\log(1+\frac{p_{m,i}h_{m,i}}{\sigma^2 d_i})} + t_i \le T, \; i = 1,2,3,...,N \\
&\quad C2: \frac{(1-\lambda_i) I_i X_i}{f_{m,i}} \le T, \; i = 1,2,3,...,N \\
&\quad C3: 0 < f_{\mathrm{m,i}} \le f_{m,i}^{\max}, \; i = 1,2,3,...,N \\
&\quad C4: \sum_{i=1}^{N} \lambda_i I_i X_i \le F \\
&\quad C5: 0 \le \lambda_i \le 1, \; i = 1,2,3,...,N \\
&\quad C6: t_i \ge 0, \; i = 1,2,3,...,N \\
&\quad C7: p_{m,i} \ge 0, \; i = 1,2,3,...,N
\end{aligned} \tag{14}$$

To be more clearly, we set $a_i = h_{m,i} / \sigma^2 d_i$, $b_i = vp_a h_{a,i}$, then, the complete expression of $E_i^{res}(f_{m,i}, p_{m,i}, \lambda_i, t_i)$ can be given by:

$$E_i^{res}(f_{m,i}, p_{m,i}, \lambda_i, t_i) = b_i t_i - (p_{m,i} \frac{\lambda_i I_i}{B\log(1+a_i p_{m,i})} + \kappa^c f_{m,i}^2 (1-\lambda_i) I_i X_i) \tag{15}$$

In P1, where the C1,C2 represent the time constraint for edge execution and local execution respectively, where the offloading has to be finished within $T - t_i$ and local execution can use up the whole time block T, C3 is the CPU frequency constraint of each mobile device, C4 is the computation ability constraint for the edge server, C5,C6 and C7 give the scope of optimization variables to be solved.

The optimal solution of problem P1 contains four solution vectors, where $\lambda=[\lambda_1,\lambda_2,...,\lambda_N]$ consists of the ratio for offloading of each mobile device, $t = [t_1, t_2, ..., t_N]$ consists of the energy-harvest time of each mobile device, $f = [f_{m,1}, f_{m,2}, ..., f_{m,N}]$ is made up with the CPU frequency of each mobile device and $p = [p_{m,1}, p_{m,2}, ..., p_{m,N}]$ is the vector constructed by the transmission power of each mobile device.

4.2 Problem Analysis

P1 is difficult to solve since that we have to obtain four unknown vectors jointly, what's more, P1 is a non-convex problem. However, through the analysis of problem P1, we can firstly transform the primary problem to a smaller dimensional one. Before giving the transform process, we first give the Lemma 1 which is the basis of the transformation.

Lemma 1 we always have $\inf_{x,y} f(x,y) = \inf_{x} \tilde{f}(x)$, where $\tilde{f}(x) = \inf_{y} f(x,y)$.

Proof: See [27].

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

According to Equation (3), it is evidently that the energy consumption of local computation increases monotonically with the increase of $f_{m,i}$, the smaller the $f_{m,i}$ is, the less energy consumed. However, the small $f_{m,i}$ will lead to the violation of constraint C2, in other words, local execution can't be finished within T. Therefore, the optimal value of $f_{m,i}$, by $f_{m,i}^{*}$ is got when using the maximum available time for computation, that is T, and thus can be given in closed-form by:

$$f_{m,i}^{*} = \frac{(1-\lambda_i) I_i X_i}{T} \tag{16}$$

by substituting the $f_{m,i}^{*}$ into P1, we can simplify the original problem P1 to P2:

$$\begin{aligned} \text{P2} \quad & \max_{\lambda,p,t} \sum_{i=1}^{N} E_i^{res}(p_{m,i}, \lambda_i, t_i) \\ s.t. \quad & C8: \frac{(1-\lambda_i) I_i X_i}{T} \le f_{m,i}^{\max} \\ & C1, C4, C5, C6, C7 \end{aligned} \tag{17}$$

where $E_i^{res}(p_{m,i}, \lambda_i, t_i) = b_i t_i - (p_{m,i} \frac{\lambda_i I_i}{B \log(1 + a_i p_{m,i})} + \kappa^c (\frac{(1-\lambda_i) I_i X_i}{T})^2 (1-\lambda_i) I_i X_i)$, C3 is transformed to C8 and C2 can be eliminated.

Next, we aim at finding the optimal $p_{m,i}$ to minimize the energy consuming. To this end, we introduce the function $g(x) = x \frac{(1-\lambda_i) I_i}{B \log(1 + a_i x)}$ to denote the relation between $p_{m,i}$ and consumed energy. It's easy to prove that the function $g(x)$ is monotone increasing, when $x \in [0, +\infty]$ and $a_i > 0$. Therefore, the smaller $p_{m,i}$ is better for consuming less energy $g(p_{m,i})$. However, the smaller transmission power will prolong the offloading time, violating the constraint C1, and hence the optimal transmission power $p_{m,i}^{*}$ can be got when use up the available offloading time $T - t_i$. The optimal $p_{m,i}^{*}$ can be given by:

$$p_{m,i}^{*} = \frac{2^{\frac{\lambda_i I_i}{B(T-t_i)}} - 1}{a_i} \tag{18}$$

by substituting the $p_{m,i}^{*}$ into P2, we can get the further simplfied problem P3:

$$\begin{aligned} \text{P3} \quad & \max_{\lambda,t} \sum_{i=1}^{N} E_i^{res}(\lambda_i, t_i) \\ s.t. \quad & C9: 0 \le t_i \le T \\ & C4, C5, C8 \end{aligned} \tag{19}$$

where $E_i^{res}(\lambda_i,t_i) = b_i t_i - ((T-t_i)\frac{2^{\frac{\lambda_i I_i}{B(T-t_i)}}}{a_i} + \kappa^c \frac{(1-\lambda_i)^3}{T^2}(I_i X_i)^3)$. Based on (18), the constraint C1 is eliminated, and accordingly the constraint C6 is transformed to C9, the constraint C7 can be eliminited.

By the two steps of substitution, the problem P1 is transformed to P3, which has the simpler mathematical formulation. Next, we will solve P3 to get the optimal combination of offloading ratio and energy harvest time, further getting the corresponding optimal transmission power and computation frequency of mobile device.

5. Problem Solution

This section will give the detail algorithm to solve P3 which is based on convex optimization and equential unconstrained minimization technique. To be convenient, we firstly do a simplified variable substitution in P3. We define $w_i = T - t_i$ in P3, and accordingly have $t_i = T - w_i$. By substituting them into P3, the detail mathematical formulation is transformed to P4 as follows:

$$\begin{aligned} \text{P4} \quad & \max_{\lambda,w} \sum_{i=1}^{N} (b_i(T-w_i) - (w_i \frac{2^{\frac{\lambda_i I_i}{B w_i}} - 1}{a_i} + \kappa^c \frac{(1-\lambda_i)^3}{T^2}(I_i X_i)^3)) \\ s.t. \quad & C3: \frac{(1-\lambda_i) I_i X_i}{T} \le f_{m,i}^{\max}, \quad i = 1,2,3,...,N \\ & C4: \sum_{i=1}^{N} \lambda_i I_i X_i \le F \\ & C5: 0 \le \lambda_i \le 1, \quad i = 1,2,3,....,N \\ & C10: 0 \le w_i \le T, \quad i = 1,2,3,...,N \end{aligned} \tag{20}$$

To prove the convexity of P4, we first give the following Lemma 2:

Lemma 2 If $f: R^n \to R$, then the perspective of f is the function $g: R^{n+1} \to R$ defined by: $g(x,t) = tf(x/t)$, dom g=$\{(x,t) | x/t \in \text{dom} f, t > 0\}$. If f is a convex function, then so is its perspective function g. Similarly, if f is concave, then so is g.

Based on Lemma 2, we defined the convex function $f(x) = d \cdot (2^{\frac{x}{c}} - 1)$, where d and c are the positive constant, and hence the second part of P4 can be denoted by $w_i f(\lambda_i / w_i)$, where the constant $c = B / I_i$ and $d = 1 / a_i$, is also convex. Besides, the third part of P4 is a one-dimensional optimization problem and can be easily proved convex in the constraint domain. Therefore, the objective function is concave and all the constraints are convex, constituting a convex optimization problem which can be optimally solved by standard convex optimization techniques.

Nevertheless, to gain engineering insights, speeding up the solving process and improving the accuracy. We firstly derive the property of optimal solution in a semi-closed form, seeking for the relationship among variables by leveraging the Lagrange method and KKT conditions, and next leveraging the property to reformulate the primary objective function to a simple polynomial form, which can be solved more easily and faster. Finally, we adopt SUMT and BFGS combining method to solve the simplified problem, and the other optimization variables can be further obtained based on the solution.

For facilitating the next analysis, we defined another function based on above defined $f(x) = 1 / a_i \cdot (2^{xI_i/B} - 1)$ as follows:

$$h(x) = f(x) - xf^{'}(x), \quad x>0 \tag{21}$$

where $f^{'}(x)$ is the first-order derivative of $f(x)$, and have the following lemma:

Lemma 3 $h(x)$ 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:

$$x^* = \frac{B}{I_i \ln 2}[W_0(\frac{a_i G + 1}{-e}) + 1].$$

The detailed processing to solve the problem P4 and further the primary problem P1 can be described as follows:

1) Finding property of optimal solution by Lagrangian method and KKT condition: Lagrangian function is usually constructed by adding all the constraints to the objective function to transform the constrained problem to unconstrained one. However, we only consider the parts of the constraints, C3, C4 in this constructing process to find the property of optimal solution. Let $\alpha > 0$ and $\beta = [\beta_1, \beta_2, \beta_3, ..., \beta_N] \succ 0$ denote the Lagrange multipliers associated with constraints C4 and C3 in problem P4, respectively. The partial Lagrange function of problem
P4 is defined as:

$$\begin{aligned} L(\lambda, w, \alpha, \beta) = \sum_{i=1}^{N} (b_i(T - w_i) - (wi \frac{2^{\frac{\lambda_i I_i}{Bw_i}} - 1}{ai} + \kappa^c \frac{(1-\lambda_i)^3}{T^2}(I_i X_i)^3) - \alpha(\sum_{i=1}^{N} \lambda_i I_i X_i - F)) \\ - \sum_{i=1}^{N} \beta_i (\frac{(1-\lambda_i) I_i X_i}{T} - f_{m,i}^{\max}) \end{aligned} \tag{22}$$

Assuming that (λ^*, t^*) denote the optimal solution for P(4) and α^*, β^* denote the optimal Lagrange multipliers. By applying the Karush-Kuhn-Tucker (KKT) conditions, the optimal solution has the following necessary and sufficient conditions:

$$\frac{\partial L(\lambda, w, \alpha, \beta)}{\partial w_i} = -b_i - h(\frac{\lambda_i^*}{w_i^*}) + \frac{\lambda_i^*}{w_i^*} h^{'}(\frac{\lambda_i^*}{w_i^*}) = 0, \quad i = 1, 2, ..., N \tag{23}$$

Based on the KKT condition (23) and Lemma 3, it can be obviously derived that the optimal solution has the following property:

$$\frac{\lambda_i^*}{w_i^*} = \frac{B}{I_i \ln 2}[W_0(\frac{-a_i b_i + 1}{-e}) + 1] \tag{24}$$

where $W_0(x)$ is principal branch of the Lambert W function defined as the solution for $W_0(x)e^{W_0(x)} = x$ [28], and the e is the base of the natural logarithm. Furthermore, we have $w_i^* = \dfrac{\lambda_i^* I_i \ln 2}{B[W_0(\frac{-a_i b_i + 1}{-e}) + 1]}$.

2) Reformulating problem by leveraging the property of optimal solution: Since P4 is proved to be convex optimization problem, and the property of optimal solution is obtained by the KKT condition. Therefore, the optimal solution of P4 will surely possess the property. By leveraging the property (24), the objective function of P4 can be transformed into a polynomial form with only one unknown vector, and the constraints can also be reduced correspondingly. Substituting (24) into P4 and getting the following problem P5:

$$\begin{aligned}
&\text{P5}\quad \max_{\lambda} \sum_{i=1}^{N} (b_i(T - \frac{\lambda_i I_i \ln 2}{B[W_0(\frac{-a_i b_i + 1}{-e}) + 1]}) - (\frac{\lambda_i I_i \ln 2}{B[W_0(\frac{-a_i b_i + 1}{-e}) + 1]} \frac{e^{W_0(\frac{-a_i b_i + 1}{-e}) + 1} - 1}{a_i} + \kappa^c \frac{(1-\lambda_i)^3}{T^2}(I_i X_i)^3)) \\
&s.t.\ \ C11:\ \max(0, 1 - \frac{Tf_{m,i}^{\max}}{I_i X_i}) \le \lambda_i \le \min(1, \frac{W_0(\frac{-a_i b_i + 1}{-e})TB}{I_i \ln 2}),\quad i = 1,2,3,\ldots,N \\
&\qquad C12:\ \sum_{i=1}^{N} \lambda_i I_i X_i \le F
\end{aligned} \tag{25}$$

where constraint C11 is derived by substituting w_i^* into C10 to transform them into constraints only on λ_i and following get the intersection of C3,C5 and C10. In problem P5, the objective function is polynomial form and the two constraints are both simple linear form. Therefore, P5 is a general non-linear constrained optimization problem and several algorithms can be applied to solve it.

3) Solving the simplified problem by SUMT and BFGS: sequential unconstrained minimization technique (SUMT) is used to reformulate the constrained problem to the unconstrained one and then solving the unconstrained optimization problem. We adopt barrier method, one of the SUMT based method, to solve P5. To leveraging the method, we defined the penalty function considering the constraints in P5 as follows:

$$M(\lambda) = \frac{1}{\sum_{i=1}^{N}(\lambda_i - \min(1, \frac{W_0(\frac{-a_i b_i + 1}{-e})TB}{I_i \ln 2})) + \sum_{i=1}^{N}(\max(0, 1 - \frac{Tf_{m,i}^{\max}}{I_i X_i}) - \lambda_i) + (F - \sum_{i=1}^{N} I_i X_i \lambda_i)} \tag{26}$$

Additionally, given the termination limits ε, the initial candidate point $\lambda^{(0)}$, the initial penalty factor μ_0, reduction coefficient of penalty factor c and the iteration counter k. We can get the unconstrained problem:

$$\min(F(\lambda)+\mu_k M(\lambda)) \tag{27}$$

where $F(\lambda)$ is defined as:

$$F(\lambda)=-\sum_{i=1}^{N}(b_i(T-\frac{\lambda_i I_i \ln 2}{B[W_0(\frac{-a_i b_i+1}{-e})+1]})-(\frac{\lambda_i I_i \ln 2}{B[W_0(\frac{-a_i b_i+1}{-e})+1]}\frac{e^{W_0(\frac{-a_i b_i+1}{-e})+1}-1}{a_i}+\kappa^c\frac{(1-\lambda_i)^3}{T^2}I_i X_i)) \tag{28}$$

We select BFGS [29] method to solve the unconstrained problem (27) in each iteration, the optimal solution denoted by $\lambda^{(k^*)}$.

The penalty factor will be update by iteration until the termination condition $\mu_k M(\lambda)<\varepsilon$ or the iteration time k is larger than 50 is satisfied, the current $\lambda^{(k^*)}$ is the optimal solution. The pseudo code of the methods is as follows:

Algorithm 1 Barrier Method

Input: $\lambda^{(0)}$: initial candidate point; μ_k: penalty factor; ϵ: termination limits; c: reduction coefficient of penalty factor; $k=0$: iteration times

1: **repeat**
2: solving the unconstrained problem (27) by BFGS method, get $\lambda^{(k^*)}$
3: $\mu_{k+1}=c\mu_k$
4: $k=k+1$
5: **until** $\mu_k M(\boldsymbol{\lambda})<\epsilon$ *or* $k=50$

Output: $\lambda^{(k^*)}$;

4) Obtaining other optimization variable: based on the optimal offloading ratio for offloading, we can get the optimal harvesting time by (24), furthermore, the transmission power and CPU frequency can be obtained by (18) and (16), respectively, and thus the primary problem P1 is solved.

Remark 1: It is likely that no solution is able to satisfy all the constraints. When it occurs, we simply give the fixed offloading ratio and harvesting time. Although this simple method will bring higher energy consumption, it still can be accepted because the failure doesn't happen very often.

Remark 2: In real environment, the task can't be partitioned as arbitrary ratio. Here, we provide a quantization method to apply the optimal offloading ratio λ. Assuming that the i-th mobile device has the possible partition set $\Omega=\{\omega_1,\omega_2,...,\omega_m\}$. For the set $\omega_i\ge\lambda_i$, we find the minimum value denoted by $\omega_{r\min}$, and for the set $\omega_i\ge\lambda_i$, also find the minimum value denoted by $\omega_{l\min}$. $\omega_{r\min}$ and $\omega_{l\min}$ are both feasible ratio close to the optimal ratio λ_i, and we choose the one that makes the residual energy maximization as the real offloading ratio.

6. Simulation Results

In this section, we present simulations to verify the advantages of proposed optimization scheme that jointly optimizing offloading ratio, harvesting time, CPU frequency and transmission power, and evaluate the performance of proposed solving algorithm. All the simulation programs are coded with Python programing language and use the PyOpt [30], a Python-based package for solving nonlinear constrained optimization problems. Besides, the simulations are running on a common PC with 3.2GHZ CPU and 6GB memory, Ubuntu 14.04 OS.

We first give the universal parameter settings for simulations. For local execution model, we set $\kappa^c = 10e^{-28}$ and the maximum CPU frequency follow the uniform distribution with $f_{m,i} \in [0.5,1]GHz$. For mobile-edge execution model, we set noise power $\sigma^2 = 10e^{-9}$, system bandwidth $B = 1MHz$, the distance between mobile device and AP follows the uniform distribution with $d_i \in [10, 20]m$. And the channel power gain from mobile device to AP is modeled as $h_{m,i} = 10^{-3} d_i^{-\alpha} \phi_i$, where ϕ_i represents the short-term fading which is assumed to be an exponentially distributed random variable with unit mean, α denotes the path-loss exponent and here set $\alpha = 2$. For energy harvest model, set $v = 0.8$, $p_a = 200W$, and the channel power gain $h_{a,i}$ from mobile device to AP is equal to $h_{m,i}$. For task model, the input size in bit and the cycles required for one-bit computation follow the uniform distribution with $I_i \in [200, 500]KB$ and $X_i \in [500, 800]$ cycles/bit, respectively. Besides, set the time-block $T = 2s$.

The settings are used in simulations in following subsections unless specified otherwise. To avoid the occasionality, the simulations are repeated 100 times for each variable value and average the result, but the other unfixed parameters are randomly selected according to the above settings every time to simulate the real environment.

The simulations in this section are to verify the advantage of proposed joint optimization scheme of offloading ratio, harvesting time, CPU frequency and transmission power, called JOS for short. For this purpose, we provide following two base methods for comparison:

1) The harvest time is fixed, half of the time block is used for energy harvest and the other half for task execution, called FHT.

2) The offloading ratio of task is fixed, half of the input bits executed on edge server and the other half executed on local device called FPR.

The first simulation shows the residual energy(J) versus the task input size(KB). By changing the task input size, we observe the changes of residual energy when dealt by the three methods respectively. Supposing that the number of mobile device $k = 10$, and the task size on all mobile devices are identical which varying from $100KB$ to $1000KB$. For each input size, we conduct 100 simulations, where d_i, X_i, $h_{m,i}$ .etc are selected randomly according to the above assumption every time. The results are shown in the **Fig. 3**.

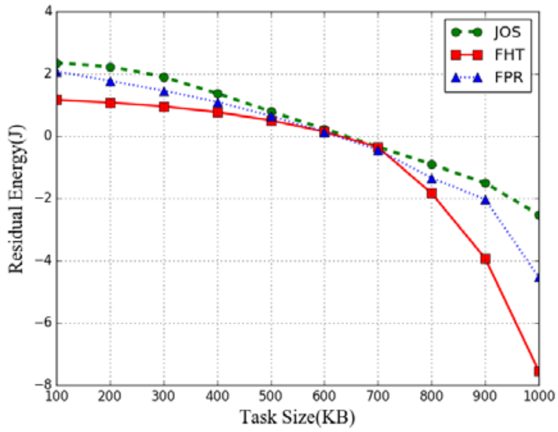


Fig. 3. Residual Energy vs Task Size

From the result in **Fig. 3**, we have the observation that the residual energy decreases with the task size increasing no matter which optimization is adopted, but proposed joint optimization scheme JOS is still superior to benchmarks. Specifically, the curve of JOS not only has the largest residual energy but also has the slowest decline rate when the task size is larger, indicating that JOS has the ability to adjust to larger tasks. This is due to that the JOS jointly optimize the offloading ratio and harvesting time to find the best combination, however, FHT and FPR are only able to find the best solution against the fixed harvesting time and offloading ratio, respectively. It is also noticed that the three schemes almost have the equal residual energy when task size is $700KB$, because of that the optimal offloading ratio and harvesting time happen to be equal with the pre-set value in FHT and FPR.

The second simulation shows the residual energy versus the transmission power of AP, observing the changes of residual energy with the transmission power increasing when dealt by the three methods. Supposing that the transmission the I_i, X_i, $h_{m,i}$ .etc are selected randomly accordingly to the above assumption. The results are show in the **Fig. 4**.

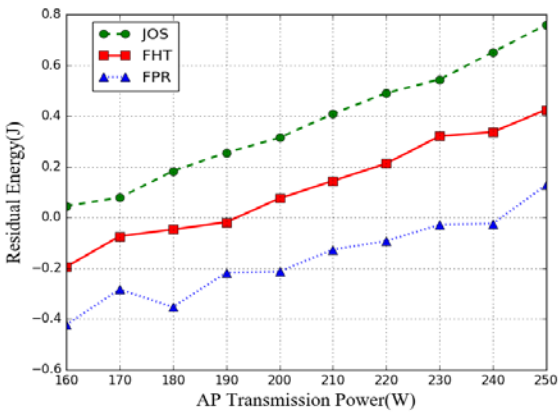


Fig. 4. Residual Energy vs AP Transmission Power

From **Fig. 4**, we obtain the observation that the JOS curve still have the largest residual energy with the AP transmission power varied from $160W$ to $250W$, indicating that proposed joint optimization scheme has the higher energy utilization rate, harvesting more energy and consuming less energy. This is because the JOS has the most suitable combination of offloading ratio and harvesting time. By comparison, the FHT has the fixed harvesting time which decides the energy amount harvested from AP, the only thing the optimization process can do is to find the best offloading ratio to finish tasks with as less as possible energy consuming in the premise of satisfying the time constraints. Similarly, the FPR has the fixed offloading ratio, which will consume much energy when the ratio is not reasonable. Besides, we have the observation that three curves are not intersected from each other and JOS curve is smoother, approximating to a linear relation. This is due to the AP transmission power only influences the amount of harvested energy, but the optimal solution is almost equal corresponding to each power, which leads to the harvested energy increases monotonically and consumed energy remains unchanged. It is also noticed that the FHT curve have more residual energy than the FPR from beginning to end, indicating that the given fixed harvesting time $1/2T$ is closer to the optimal value, while the given offloading ratio has bigger difference from optimal value.

The third simulation shows the length of time-block T versus failure ratio which is defined that the task can't be completed within T. Supposing that the time block varies from $1.5s$ to $2.4s$. With the T increasing, the changes of failure rate are shown in **Fig. 5**.

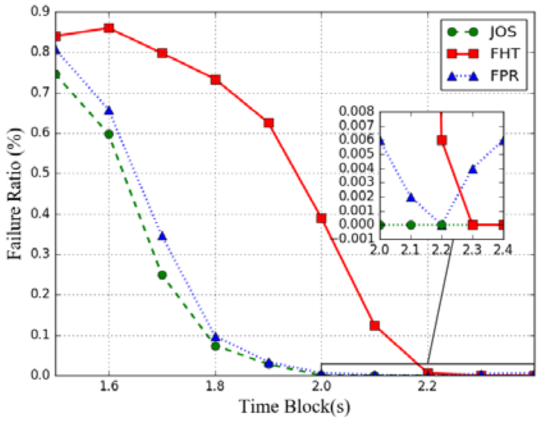


Fig. 5. Failure Ratio vs Time Block

We have several observations from **Fig. 5**. First, the failure ratio is quite high when the time block is smaller, but proposed JOS have relatively lower failure ratio, because JOS has the ability to find the optimal combination of offloading ratio and harvesting time to find the most suitable combination for given time constraint, but the benchmarks can only optimize one of them. Second, with the increasing of time block, all the three curves decrease rapidly, especially the JOS and FPR, and the FPR curve almost has the equal failure ratio with JOS, but still little higher, indicating that the FPR is usually able to find the suitable time division to make sure tasks finished for given offloading ratio. Besides, from the the magnified part of **Fig. 5**, we can observe that FPR isn't stable, still having failure task even if the time block is large enough, but JOS is able to make sure the completion. This is due to that it's possible that FPR can't find the reasonable time division in some cases for given offloading ratio, but JOS avoids the cases by joint optimization.

6.2 Performance of Algorithm

The simulations in this section are to verify the performance of proposed algorithm on solving the REM problem, including computation time and the effect of optimal solution. To this end, we select the following two methods as the benchmarks.

1) Augmented Lagrangian methods with general lower-level constraints (ALGENCAN): It solves the general non-linear constrained optimization problem without resorting to the use of matrix manipulations [31]. We adopt ALGENCAN method to solve P4 directly, which isn't simplified by the KKT conditions.

2) Sequential Least Squares Programming(SLSQP): It is a sequential least squares programming algorithm which uses the Han-Powell quasi-Newton method with a BFGS update of the BCmatrix and an L1-test function in the step-length algorithm [32]. We adopt SLSQP method to solve the simplified problem P5.

In this simulation, we test the time consumption of three methods on solving the REM problem, observing the changes of computation time and the effect of optimal solution on maximize the residual energy with the number of user increasing. Proposed convex optimization and SUMT combination method is short for COSUMT in below simulation. The results are shown in **Fig. 6**:

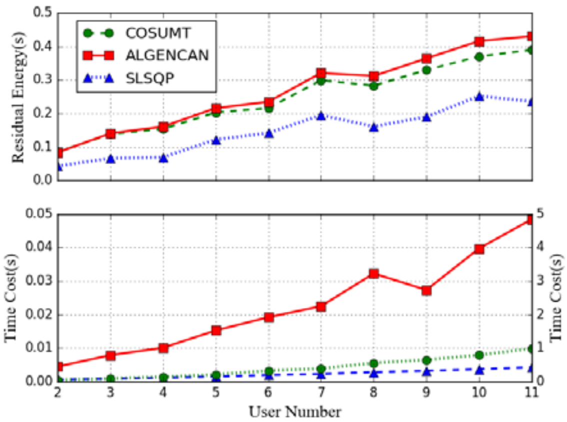


Fig. 6. Performance vs User Number

The **Fig. 6** shows the performance of three methods on residual energy and computation time simultaneously. We can observe from the above section that the residual energy increase with the user number increasing, and our proposed COSUMT method has much better objective value than the SLSQP method and closer to the ALGENCAN method. However, from the below section we can see that our proposed COSUMT method has closer computation time to SLSQP, but much shorter than ALGENCAN method, which is even thousands of times longer than COSUMT and hence shown at the right y axis. This is because our proposed method has simplified the original problem by convex optimization theory, simplifying the primary problem in negligible time. This figure proves that proposed COSUMT algorithm is able to get an approximate optimal solution with less time cost.

7. Conclusion

In this paper, we propose a MEC and EH integrated system with partial offloading task. Mobile devices under the system are able to harvest the energy and then execute tasks on edge server and local device simultaneously. By jointly optimizing the offloading ratio, energy harvesting time, the CPU frequency and transmission power of mobile device, we aim at solving the REM and ECM problem at the same time to prolong the battery life. For solving the problem, we leverage the combination of convex optimization and SUMT method, which aiming at obtaining lower time overhead and higher accuracy. Simulation result shows that, our jointly optimizing scheme outperforms the conventional single optimization, and proposed algorithm proved to be more efficiency. MEC and EH integrated system will be the hotpot research point because of its practicability, but the related researches are still at the initial stage. Our work provided a new idea on for related researches. Next, we will explore the optimal framework of EH and MEC integrated system, and leverage more efficient algorithm to build a energy-efficient MEC system.

References

[1] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications & Mobile Computing*, vol. 13, no. 18, pp. 1587-1611, 2013. Article (CrossRef Link)

[2] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5g heterogeneous networks," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 45–55, 2014. Article (CrossRef Link)

[3] ETSI, "Etsi mobile-edge computing - introductory technical white paper," 2016.

[4] M. T. Beck, S. Feld, C. Linnhoff-Popien, and U. Ptzschler, "Mobile edge computing," *Informatik-Spektrum*, vol. 39, no. 2, pp. 1–7, 2016. Article (CrossRef Link)

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," 2017.

[6] X. Lu, P. Wang, D. Niyato, I. K. Dong, and Z. Han, "Wireless networks with rf energy harvesting: A contemporary survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 757–789, 2014. Article (CrossRef Link)

[7] S. Bi, C. K. Ho, and R. Zhang, "Wireless powered communication: opportunities and challenges," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 117–125, 2014. Article (CrossRef Link)

[8] K. Kumar and Y. H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?," *Computer*, vol. 43, no. 4, pp. 51–56, 2010. Article (CrossRef Link)

[9] S. Kosta, A. Aucinas, P. Hui, and R. Mortier, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. of INFOCOM, 2012 Proceedings IEEE*, pp. 945–953, 2012. Article (CrossRef Link)

[10] E. Cuervo, A. Balasubramanian, D. K. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui:making smartphones last longer with code offload," in *Proc. of International Conference on Mobile Systems, Applications,and Services*, pp. 49–62, 2010. Article (CrossRef Link)

[11] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proc. of Conference on Computer Systems*, pp. 301–314, 2011. Article (CrossRef Link)

[12] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, pp. 1–1, 2017. Article (CrossRef Link)

[13] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Joint allocation of computation and communication resources in multiuser mobile cloud computing," in *Proc. of IEEE 14th Workshop on Signal Processing Advances in Wireless Communications*, vol. 395, no. 6, pp. 26–30, 2013. Article (CrossRef Link)

[14] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading sing dynamic voltage scaling," *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 4268–4282,2016. Article (CrossRef Link)

[15] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal & Information Processing Over Networks*, vol. 1, no. 2, pp. 89–103, 2014. Article (CrossRef Link)

[16] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2017. Article (CrossRef Link)

[17] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Transactions on Wireless Communications*,vol. 16, no. 9, pp. 5994–6009, 2017. Article (CrossRef Link)

[18] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1757–1771, 2016. Article (CrossRef Link)

[19] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2017. Article (CrossRef Link)

[20] 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. Article (CrossRef Link)

[21] S. Bi and Y. J. A. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," 2017. Article (CrossRef Link)

[22] H. Gao, W. Ejaz, and M. Jo, "Cooperative wireless energy harvesting and spectrum sharing in 5g networks," *IEEE Access*, vol. 4, pp. 3647–3658, 2017. Article (CrossRef Link)

[23] L. Yang, J. Cao, S. Tang, T. Li, and A. T. S. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," in *Proc. of IEEE International Conference on Cloud Computing*, pp. 23–32, 2012. Article (CrossRef Link)

[24] O. Muoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Transactions on Vehicular Technology*,vol. 64, no. 10, pp. 4738–4755, 2014. Article (CrossRef Link)

[25] Jan M. Rabaey, "Digital integrated circuits: a design perspective," Prentice Hall, 1996.

[26] G. Qu, "What is the limit of energy saving by dynamic voltage scaling?," in *Proc. of Ieee/acm International Conference on Computer Aided Design*, pp. 560–563, 2001. Article (CrossRef Link)

[27] S. Boyd and L. Vandenberghe, "Convex Optimization," *Cambrige University Press* 2004.

[28] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the lambert ww function.," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

[29] Broyden-Fletcher–Goldfarb–Shanno algorithm, Article (CrossRef Link).

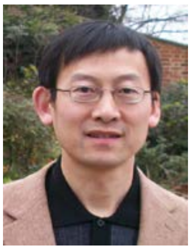
[30] R. E. Perez, P. W. Jansen, and J. R. R. A. Martins, "pyopt: a python-based object-oriented framework for nonlinear constrained optimization," *Structural & Multidisciplinary Optimization*, vol. 45, no. 1, pp. 101-118, 2012.

[31] R. Andreani, E. G. Birgin, J. M. Mart&Iacute, and N. L. Schuverdt, "On augmented lagrangian methods with general lower-level constraints," *Siam Journal on Optimization*, vol. 18, no. 4, pp. 1286–1309, 2007.

[32] D. Kraft, "A software package for sequential quadratic programming," 1988.

Peng Liu received the MS degrees from the College of Computer Science and Technology, Jilin University in 2013. Currently, he is PhD candidate at the College of Computer Science and Technology, Jilin University, China. His main research interests include mobile cloud computing, Mobile edge computing, SDN.

Gaochao Xu received the BS, MS, and PhD degrees from the College of Computer Science and Technology, Jilin University in 1988, 1991,and 1995, respectively. Currently, he is the professor and PhD supervisor at the College of Computer Science and Technology, Jilin University, China. His main research interests include distributed system, grid computing, cloud computing, Internet of things, information security, software testing, and software reliability assessment, etc. As a person in charge or a principal participant, he has finished more than 10 national, provincial, and ministerial level research projects of China.



Kun Yang received the PhD degree from the Department of Electronic and Electrical Engineering, University College London (UCL), United Kingdom. He is currently a full professor and the head of Network Convergence Laboratory (NCL) in the School of Computer Science and Electronic Engineering, University of Essex, United Kingdom. Before joining in the University of Essex at 2003, he worked at UCL on several European Union Research Projects such as FAIN, MANTRIP, and CONTEXT. His current major research interests include heterogeneous wireless networks, fixed mobile convergence, and cloud computing. He has published more than 150 journal and conference papers in the above areas. He serves on the editorial boards of both IEEE and non-IEEE journals. He is a senior member of the IEEE and a fellow of the IET.

Kezhi Wang received his B.E. and M.E. degrees from the College of Automation, Chongqing University, P.R.China, in 2008 and 2011, respectively. He received his Ph.D. degree from the University of Warwick, United Kingdom, in 2015. He is currently a senior research officer at the University of Essex, United Kingdom. His research interests include wireless communication, signal processing, and mobile cloud computing.

Yang Li received the MS degrees from the College of Computer Science and Technology, Jilin University in 2014. Currently, he is PhD candidate at the College of Computer Science and Technology, Jilin University, China. His main research interests include mobile cloud computing, distributed computing.