Energy-Efficient Joint Computation Offloading and Resource Allocation in Multi-User MEC Systems

Yao Meng, Janxin Dai

College of Telecommunications and Information Engineering, Nanjing University of Post and Telecommunications, Nanjing 210003, China.

College of Science, Nanjing University of Post and Telecommunications, Nanjing 210003, China.

E-mail: daijx@njupt.edu.cn

Abstract. Mobile edge computing (MEC) is envisioned as an emerging paradigm to enable energy-constrained and computation-limited user equipments (UEs) to offload various computation tasks to the edge of mobile networks in order to save energy consumption and prolong the battery life of UEs. In this paper, we consider a multi-user MEC system where each UE has a computation task to be processed locally or offloaded for remote processing. Specifically, we investigate the task offloading problem with the aim of minimizing the energy consumption of UEs via jointly optimizing the task admission decision, the transmission power, local computing and edge computing capacities. To solve this nonconvex problem, we transform it into four subproblems and propose a low-complexity algorithm by solving subproblems in an iterative manner. Simulation results demonstrate that the proposed algorithm requires lower energy consumption than the existing benchmark schemes.

1. Introduction

With the popularity of resource intensive mobile applications, such as virtual reality services and high quality video playing, user equipments (UEs) are experiencing very hard time, as they normally are resource constrained and may not be able to complete the above mentioned tasks in required time. Fortunately, mobile edge computing (MEC) has been proposed to enable UEs to offload computations to the edge servers [1].

Through offloading, UEs cannot only prolong battery life and reduce their energy consumption but also save on-board resource, such as storage and CPU. In the MEC framework, there are generally two types of computation offloading: binary offloading and partial offloading. In partial offloading [2][3], tasks can be divided into sub-tasks which can be executed at MEC servers or locally. While in the binary offloading [4][5], tasks are atomic and cannot be split into sub-tasks for local execution or edge execution.

The offloading decision is normally very difficult to solve as it involves integer variables and is normally NP-hard, especially in the situation where there is large number of users and they have delay-sensitive tasks. This problem becomes more challenging if we also deal with the transmission power and computational capacity allocation problems jointly. In fact, the offloading decision is highly coupled with the power and computational resource allocation. Addressing the above problems successfully can lead to significant energy reduction for the users.
This paper introduces a framework for joint computation offloading and resource allocation in a multi-user MEC system that consists of a base station (BS) and multiple single-antenna UEs. We consider a joint optimization of task offloading decision, transmission power, local computing and edge computing capacities that minimizes the energy consumption of UEs. We assume that every task cannot be split for partial local computing and partial offloading. In order to solve this nonconvex problem, we propose a four-stage scheme by decomposing the original problem into four subproblems: We first decide whether UEs should offload the tasks or execute it locally. Then based on the determined task offloading decision, we derive closed-form solutions of the optimal transmission power and optimal local computing capability. The optimal edge computing capability can be obtained by using a bisection search algorithm. We solve these subproblems and propose a low-complexity algorithm by solving subproblems iteratively. Numerical results show that the proposed algorithm can significantly reduce the energy consumption of UEs compared with the existing benchmark schemes without such a joint consideration.

[6] considered the task offloading problem in a multi-user proximate cloud by jointly optimizing the offloading decision and resource utilization to maximize system utility. A heuristic offloading decision algorithm which is semi-distributed and was proposed to decide whether to send offloading request and form a local optimal offloading set according to the priority order. [7] formulated the joint optimization problem to minimize a weighted sum of costs of energy computation and the maximum delay among all users as a mixed-integer programming. A three-step heuristic algorithm and a novel randomization mapping method were proposed to solve the optimization efficiently.

Despite the same objective as in [5], our work is different from [5] in that we jointly optimize task offloading decision, transmission power, local computing and edge computing capacities while [5] only optimized admission decisions and allocated computational resources. Apart from the algorithms of Local-Only and Offload-Only, [5] adopted Branch and Bound and Partial offloading for comparison purpose. These two algorithms are not applicable in our problem since there are more involved optimization variables and the objective function is not linear.

The rest of the paper is organized as follows. Section 2 introduces the system model of a multi-user MEC system and formulates the optimization problem. A joint optimization algorithm is proposed in Section 3. In Section 4, simulation results are shown to evaluate the performance of the proposed algorithm compared with other algorithms, followed by the conclusion in Section 5.

2. System Model
In this section, we introduce system model and formulate the optimization problem, which aims to minimize the energy consumption of UEs under the latency constrains of the computational tasks.
As shown in Fig. 1, we consider a multi-user MEC system where there is a BS that is equipped with a MEC server. In this system, the set of all UEs which are connected to the BS are denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$. Each UE has a computation task to be completed locally at the UEs or offloaded to the MEC server.

In this paper, we define $a_i$ as the task offloading decision which means that the possible place that the $i$-th UE’s task can be executed, where $a_i = 0$ denotes that UE processes the task locally, while $a_i = 1$ denotes that UE offloads the task to the MEC server. Then we have $C1$: $a_i = \{0, 1\}$, $\forall i \in \mathcal{N}$ (1). Similar to [8], the task $U_i$ can be defined by a triplet $U_i = \{F_i, D_i, T\}$, $\forall i \in \mathcal{N}$, where $F_i$ specifies the total number of the CPU cycles required to accomplish $U_i$, $D_i$ denotes the data size transmitting to the MEC server if the task chooses to offload and $T$ is the maximum latency constraint by this task. In addition, $D_i$ and $F_i$ can be obtained by using the techniques presented in [9].

Let $f_i$ denote the computational resources allocated to $i$-th UE at the MEC server, thus we can obtain remote execution time $T_i^e$ as $T_i^e = F_i / f_i$. The time to offload the data is given by $T_i^d = D_i / r_i$, where $r_i$ is data rate of device $i$. Assume that the channel follows Rayleigh distribution which is
denoted as \( h_i \), then \( r_i \) can be calculated as \( r_i = B \log_2 \left( 1 + \frac{p_i^T}{h_i} \sigma^2 \right) \), where \( B \) is channel bandwidth, \( \sigma^2 \) is noise power and \( p_i^T \) is the transmitting power from \( i \)-th UE to the MEC server. If the task chooses to offload to the server, it must meet the latency requirement as \( D_i / r_i + F_i / f_i \leq T, \forall i \in \mathcal{N} \) (2). The total computation capacity for the MEC server is constrained by

\[
C2: f_{\text{total}} = \sum_{i=1}^{\mathcal{N}} a_i f_i \leq f_{\text{max}}, \forall i \in \mathcal{N}.
\]

For local computing, let \( f_i^l \) and \( T_i^l \) denote the local computing capability of the device \( i \) and the time for local execution respectively, as given by \( T_i^l = F_i / f_i^l \). If the task chooses to execute in UE itself, we have \( F_i / f_i^l \leq T \). Hence we can have \( C3: a_i \left( D_i / r_i + F_i / f_i \right) + (1-a_i) F_i / f_i^l \leq T \). The computation capacity for UEs is constrained by

\[
C4: (1-a_i) f_i^l \leq f_{i,\text{max}}, i \in \mathcal{N} \quad (3).
\]

The energy consumption of \( i \)-th UE can be denoted by

\[
E_{\text{UE}}^i = \begin{cases} p_i^T T_i^l & \text{if offloading} \\ p_i^E T_i^l & \text{if local execution} \end{cases}
\]

where \( p_i^E \) is \( i \)-th UE’s local computing power and \( p_i^E = k_i \left( f_i^l \right)^\nu \), where \( k_i \geq 0 \) is the effective switched capacitance and \( \nu_i \geq 1 \) is the positive constant. Typically we set \( k_i = 10^{-13} \) and \( \nu_i = 3 \). The UE power is constrained by

\[
C5: a_i p_i^E + (1-a_i) k_i \left( f_i^l \right)^\nu \leq P_{i,\text{max}}, i \in \mathcal{N} \quad (5).
\]

To minimize the energy consumption of UEs, then we have the optimization problem as follows:

\[
P: \min_{a_i, f_i, \forall i \in \mathcal{N}} \sum_{i=1}^{\mathcal{N}} \left[ a_i p_i^T D_i / r_i + (1-a_i) k_i \left( f_i^l \right)^2 F_i \right]
\]

s.t. \( C1: a_i = \{0,1\}, i \in \mathcal{N} \)

\[
C2: f_{\text{total}} = \sum_{i=1}^{\mathcal{N}} a_i f_i \leq f_{\text{max}}, i \in \mathcal{N}
\]

\[
C3: a_i \left( D_i / r_i + F_i / f_i \right) + (1-a_i) F_i / f_i^l \leq T, i \in \mathcal{N}
\]

\[
C4: (1-a_i) f_i^l \leq f_{i,\text{max}}, i \in \mathcal{N}
\]

\[
C5: a_i p_i^T + (1-a_i) k_i \left( f_i^l \right)^\nu \leq P_{i,\text{max}}, i \in \mathcal{N}
\]

Here, \( C1 \) means that the task \( U_i \) can be either executed locally or offloaded to the edge servers. \( C2 \) ensures that the total computational resource at the MEC server is no more than the maximum computational capability. All UEs must satisfy the latency requirement given in \( C3 \). \( C4 \) ensures that the computational resource assigned to UEs cannot exceed the maximum computational capability. \( C5 \) represents that the transmission power of UE is less than the maximum transmission power.

3. Proposed Algorithm

Note that the task admission decision \( a_i \) is binary and the objective function and constrains are nonconvex, Problem \( P \) is obviously a nonconvex problem. It is challenging to solve this nonconvex problem to obtain a globally optimal solution. Instead, we propose a joint optimization method to solve \( P \) efficiently.
3.1. Offloading decision
Given $f_i$, $f_i'$, and $p_i^T$, the optimization problem is still non-convex due to the integer requirement associated with $a_i$. To resolve this issue, we relax $a_i \in [0,1], i \in \mathcal{N}$. Denoting $\lambda_i$, $\mu_i$, $\eta_i$, and $\xi_i$ as the Lagrange multipliers, then we can obtain the Lagrangian function of (6) as

$$L(\lambda_i, \mu_i, \eta_i, \xi_i) = \min_a \sum_{i \in \mathcal{N}} \left( a_i p_i^T \frac{D_i}{r_i} + (1-a_i) k_i \left( f_i^T \right)^2 F_i \right) + \sum_{i \in \mathcal{N}} \lambda_i \left( a_i \left( \frac{D_i}{r_i} + \frac{F_i}{f_i} \right) + (1-a_i) \frac{F_i}{f_i} - T \right)$$

$$+ \sum_{i \in \mathcal{N}} \mu_i \left( (1-a_i) f_i' - f_{i,\text{max}} \right) + \sum_{i \in \mathcal{N}} \eta_i \left( a_i p_i^T + (1-a_i) k_i \left( f_i' \right)^3 - P_{i,\text{UE}} \right) + \xi_i \left( \sum_{i=1}^N a_i f_i - f_{\text{max}} \right).$$

Through proper simplification we can obtain

$$a_i^* = \begin{cases} 1 & \theta(\lambda_i, \mu_i, \eta_i, \xi_i) < 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\theta(\lambda_i, \mu_i, \eta_i, \xi_i) = p_i^T D_i / r_i - k_i \left( f_i' \right)^2 F_i + \lambda_i \left( D_i / r_i + F_i / f_i \right) - \lambda_i F_i / f_i' - \mu_i f_i' + \eta_i p_i^T - \eta_i k_i \left( f_i' \right)^3 + \xi_i f_i$.

Therefore, we can obtain the offloading set. Let $\mathcal{N}_0 = \left\{ i \mid a_i^* = 0 \right\}$ denotes the set of UE which choose to execute the task locally and $\mathcal{N}_1 = \left\{ i \mid a_i^* = 1 \right\}$ denotes the set of UE which decide to offload the task. The value of $\lambda_i$, $\mu_i$, $\eta_i$ and $\xi_i$ can be updated by the sub-gradient method, which is given by

$$\lambda_{i+1} = \lambda_i + \epsilon \left[ a_i \left( \frac{D_i}{p_i^T} + \frac{F_i}{f_i} \right) + (1-a_i) \frac{F_i}{f_i} - T \right] \mathbf{i} \in \mathcal{N}$$

$$\mu_{i+1} = \mu_i + \epsilon \left( (1-a_i) f_i' - f_{i,\text{max}} \right) \mathbf{i} \in \mathcal{N}$$

$$\eta_{i+1} = \eta_i + \epsilon \left( a_i p_i^T + (1-a_i) k_i \left( f_i' \right)^3 - P_{i,\text{UE}} \right) \mathbf{i} \in \mathcal{N}$$

$$\xi_{i+1} = \xi_i + \epsilon \left( \sum_{i=1}^N a_i f_i - f_{\text{max}} \right) \mathbf{i} \in \mathcal{N}$$

where $\epsilon$ is step-size sequence.

According to (8) and (9), the complexity of UEs classification is $\mathcal{O}(\mathcal{N})$, and the complexity of updating Lagrange multipliers is also $\mathcal{O}(\mathcal{N})$.

3.2. Optimal transmitting power $p_i^T$

We can optimize $p_i^T$ when the UEs choose to offload where $a_i$ is equal to one. To solve $\mathcal{P}$ with given $f_i$ and $f_i'$, the power control problem can be formulated as

$$\mathcal{P1}: \min_{p_i} \sum_{i \in \mathcal{N}_1} p_i^T \frac{D_i}{r_i}$$

s.t. $C3: \frac{D_i}{r_i} + \frac{F_i}{f_i} \leq T, i \in \mathcal{N}_1$ (10).

$C5: p_i^T \leq P_{i,\text{UE}}, i \in \mathcal{N}_1$

By substituting $r_i$ into $C3$ and combining with $C5$, we can obtain the lower bound and the upper bound of $p_i^T$. By checking the first-order derivative of the objective function with respect to $p_i^T$, we
can prove that the objective function is an increasing function in $p_i^T$. Then, given $f_i$, the optimal $p_i^{*T}$ can be derived as

$$p_i^{*T} = \frac{\sigma^2}{h_i} \left( \frac{D_i}{2 \left( \frac{h}{T_f} \right) - 1} \right)$$

(11).

3.3. Optimal local computing capability $f_i^l$

When these UEs belong to $N_0$, the original objective function can be rewritten as

$$\mathcal{P}2: \min_{f_i^l} \sum_{i \in N_0} k_i \left( f_i^l \right)^2 \cdot F_i$$

s.t.

$$C3: \frac{F_i}{f_i^l} \leq T, i \in N_0$$

(12).

$$C4: f_i^l \leq f_{i,\max}, i \in N_0$$

$$C5: k_i \left( f_i^l \right)^3 \leq P_{i,\max}, i \in N_0$$

Based on C3, C4 and C5, the lower bound and the upper bound of $f_i^l$ can be obtained. In addition, the objective function of $\mathcal{P}2$ increases with $f_i^l$, then the optimal solution is given by

$$f_i^{*l} = \frac{F_i}{T}$$

(13).

3.4. Optimal edge computing capability $f_i$

By substituting the optimal power $p_i^{*T}$ and offloading decision $a_i^*$ into the original objective function, we can obtain

$$\mathcal{P}3: \min_{F_i} \sum_{i \in N_1} \frac{\sigma^2}{h_i} \left( \frac{D_i}{2 \left( \frac{h}{T_f} \right) - 1} \right) \left( T - \frac{F_i}{f_i} \right)$$

s.t.

$$C2: f_{\text{total}} = \sum_{i \in N_1} f_i \leq f_{\text{max}}$$

(14).

$$C3: \frac{D_i}{r_i} + \frac{F_i}{f_i} \leq T, i \in N_1$$

According to the concavity rules of composite functions, the objective function of $\mathcal{P}3$ is a convex function. Thus $\mathcal{P}3$ is convex.

Denoting $\zeta$ as the Lagrange multiplier, the Lagrangian function of $\mathcal{P}3$ is

$$\mathcal{L} = \sum_{i \in N_1} \frac{\sigma^2}{h_i} \left( \frac{D_i}{2 \left( \frac{h}{T_f} \right) - 1} \right) \left( T - \frac{F_i}{f_i} \right) + \zeta f_i - \zeta f_{\text{max}}$$

(15). The Karush-Kuhn-Tucker(KKT) conditions of $\mathcal{P}3$ are
\[
\frac{\partial L}{\partial f_i} = 2 \sigma^2 \left( \frac{D_i f_i}{h_i} \right) - D_i f_i \left( T - F_i \right) + \sigma^2 \left( \frac{D_i}{h_i} \right) \left( \frac{B(T_f - F_i)}{f_i} \right)^2 - 1 \frac{F_i}{f_i^2} + \zeta = 0
\]  

(16).

If \( \zeta = 0 \), through proper simplification, we can have
\[
\ln 2 * 2 \frac{D_i f_i}{h_i} \left( T - F_i \right) + \frac{D_i}{h_i} \left( \frac{B(T_f - F_i)}{f_i} \right)^2 - 1 \frac{F_i}{f_i^2} + \zeta = 0
\]  
(17). By defining \( \mathcal{A} = \frac{D_i f_i}{h_i} \left( T - F_i \right) \) and applying the exponential operation at both sides, the equation can be re-expressed as \( e^{\ln 2} \mathcal{A} \left( T - F_i \right) = -e^{-1} \). According to the definition and property of Lambert function, we have \( \mathcal{A} \left( T - F_i \right) = -e^{-1} \), then we can obtain \( \mathcal{A} = 0 \), but this result has no physical meaning.

If \( \zeta > 0 \), we can obtain
\[
\ln 2 \sigma^2 \left( \frac{D_i f_i}{h_i} \right) - \frac{D_i f_i}{h_i} \left( T - F_i \right) + \frac{D_i}{h_i} \left( \frac{B(T_f - F_i)}{f_i} \right)^2 - 1 \frac{F_i}{f_i^2} + \zeta = 0
\]  
(18).

The left side of (18) is the first derivative of the Lagrangian function (15) and the positive constant \( \zeta \).

The first derivative of the Lagrangian function is the second derivative of the objective function (14).

Since the objective function is a convex function, the first derivative on the left side of (18) is greater than 0. So the left side of (18) is monotonically increasing. We can solve this problem by using bisection search method.

3.5. **Proposed Iterative Algorithm**

In this section, we solve \( \mathcal{P} \) by the proposed iterative algorithm as shown in Algorithm 1. The iterative algorithm 1 always converges. Denote \( T(a_i, f_i, f_i', p_i') \) as the objective values of \( \mathcal{P} \). First, with fixed \( (f_i', f_{i-1}', p_{i-1}') \), the optimal offloading decision \( a_i \) is obtained. Then it follows that
\[
V_{obj} = T(a_{i-1}, f_{i-1}, f_{i-1}', p_{i-1}')
\]  
(19).

Next, with fixed \( (a_i, f_{i-1}, p_{i-1}') \), \( f_i' \) is the optimal local computing capability. The optimal \( f_i \) is obtained with given \( (a_i, f_i', p_i') \). And the optimal \( p_i' \) is obtained with fixed \( (a_i, f_i', f_i) \). Since \( p_i' \), \( f_i \), \( f_i' \) are the globally optimal solutions, we have
which shows that the objective function of (6) is non-increasing when variables are updated. Since the sum energy consumption is always positive and lower-bounded by zero, the proposed algorithm always converges.

Algorithm 1: Proposed Iterative Algorithm for (6):
1. Set the tolerance \( \varepsilon \) and the iteration number \( t = 0 \). Initialize the initial solution \((a_0, f_0^l, f_0^r, p_0^r)\).
2. Compute the objective function according to (6) and denote it as \( V_{obj}^0 = T(a_0, f_0^l, f_0^r, p_0^r) \).
3. Repeat
   4. \( t = t + 1 \).
5. Obtain the optimal \( a_t^* \) and offloading sets according to (8) with fixed \((f_t^l, f_t^r, p_t^r)\).
6. Obtain the optimal \( f_t^{ir} \) of \( \mathcal{P}2 \) with fixed \((a_t^*, f_t^r, p_t^r)\).
7. Obtain the optimal \( f_t^{ir} \) of \( \mathcal{P}3 \) with given \((a_t^*, f_t^{ir}, p_t^r)\).
8. Obtain the optimal \( p_t^{ir} \) of \( \mathcal{P}1 \) with fixed \((a_t^*, f_t^{ir}, f_t^{ir})\).
9. Compute the objective function and denote it as \( V_{obj}^t \).
10. Until \( |V_{obj}^t - V_{obj}^{t-1}| / V_{obj}^{t-1} < \varepsilon \).

4. Simulation Results
In this section, some numerical results are presented to evaluate the proposed joint algorithm and analyse the performance. We consider a network where there is a base station that is equipped with a MEC server and the number of UEs connecting to the BS is 5. We adopt the channel that follows the Rayleigh distribution. The channel bandwidth is \( B = 1 \text{MHz} \). The noise power is \( \sigma^2 = -110\text{dBm} \) and \( h_t = 10^{-3} \). For each UEs, we set \( D = 10^6 \text{bits} \), \( F = 1 \text{M CPU cycles} \), \( T = 1 \text{s} \), the maximal transmission power is \( P_{i,max}^{UE} = 17\text{dBm} \) and the maximal computation capacity is \( f_{i,max} = 10^8 \text{cycles/s} \). For the MEC server, the total computation capacity for the MEC server is \( f_{max} = 10^9 \text{cycles/s} \).

In addition to the proposed joint algorithm, we also take the following algorithms into consideration for comparison. One is local execution (Local-Only) which all tasks execute locally without offloading. The other is MEC execution (Offload-Only) which all tasks are offloaded to the MEC server for remote execution. The other is the Exhaustive method (Exhaustive method) which involves all combinations of \( a_i \).
In Fig. 2, we illustrate the sum energy consumption on UEs versus the total number of CPU cycles $F$. From this figure, we find that the sum energy consumption increases with total number of the CPU cycles while the offload-only scheme remains almost unchanged. Since large number of the CPU cycles requires more local and edge computing resources allocated to tasks to meet strict constraints which results in high energy consumption. When the total number of CPU cycles becomes larger, the proposed joint algorithm outperforms other schemes obviously.

We show the sum energy consumption on UEs versus the maximal latency $T$ in Fig. 3. The proposed joint algorithm performs better than other two schemes. It is observed that the energy consumption with the offload-only scheme remains almost unchanged. It is due to the fact that the
optimal time is around 500ms. By contrast, the energy consumption with other three schemes decreases monotonically as time increases. When $T$ is about 1020ms, the local-only scheme consumes less energy than the offload-only scheme because when $T$ increases the local computing capability becomes small.

![Fig.4 The sum energy consumption on UEs versus the total number of CPU cycles $D$.](image)

The sum energy consumption of UEs versus the data size $D$ is depicted in Fig.4. It is shown that the energy consumption by all schemes increases as data size becomes large and finally becomes stable. It is due to the fact that more transmission power should be consumed to compute tasks. The offload-only scheme performs better than the local-only scheme which indicates that it is a better choice to offload to the MEC server for edge execution especially when $D$ increases. We can also find that the proposed joint algorithm consumes lower energy compared with other schemes.
Fig. 5 shows the sum energy consumption on UEs versus the bandwidth $B$. It is observed that the sum energy consumption by three schemes increases while the other scheme remains unchanged. Therefore, when the system bandwidth is small, offloading all the computation tasks to the MEC server is not efficient. We can also find that the performance of the offloading only scheme becomes better than the local only scheme as the system bandwidth is large enough. We can again observe that the proposed joint algorithm generates significantly lower overhead compared to the other schemes.

5. Conclusion
In this paper, we investigated the computation offloading and resource allocation problem in a multi-user MEC system. To minimize the energy consumption of the UEs under several constrains, we formulated this problem as a constrained optimization to jointly optimize the task admission decision, the transmission power, local computing and edge computing capacities. In this regard, a low-complexity algorithm was proposed to solve this nonconvex problem. We decomposed the original problem into four subproblems which can be proved to be convex problems and can obtain the optimal solutions. Then we conducted simulation to analyze the performance of our proposed joint design and the other two algorithms under different values of key parameters. The simulation results indicated that the proposed design outperforms the other two algorithms and can obtain lower energy consumption.

References
[1] P. Mach and Z. Becvar. Mobile edge computing: A survey on architecture and computation offloading[J]. IEEE Communications Surveys Tutorials, 2017, 19(3): 1628-1656.
[2] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief. Delay-optimal computation task scheduling for mobile-edge computing systems[J]. IEEE International Symposium on Information Theory 2016, 1451–1455.
[3] Y. Mao, J. Zhang, and K. B. Letaief. Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems[J]. IEEE Wireless Communications and Networking Conference, 2017, 1–6.
[4] S. Bi and Y. J. Zhang. Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading[J]. IEEE Transactions on Wireless Communications, 2018, 17(6):4177–4190.

[5] X. Lyu, H. Tian, W. Ni, Y. Zhang, P. Zhang, and R. P. Liu. Energy-efficient admission of delay-sensitive tasks for mobile edge computing[J]. IEEE Transactions on Communications, 2018, 66(6):2603–2616.

[6] X. Lyu, H. Tian, C. Sengul and P. Zhang. Multiuser Joint Task Offloading and Resource Optimization in Proximate Clouds[J]. IEEE Transactions on Vehicular Technology, 2017, 66(4):3435-3447.

[7] M. Chen, M. Dong and B. Liang. Resource Sharing of a Computing Access Point for Multi-User Mobile Cloud Offloading with Delay Constraints[J]. IEEE Transactions on Mobile Computing, 2018, 17(12):2868-2881.

[8] K. Wang, K. Yang, and C. Magurawalage. Joint energy minimization and resource allocation in C-RAN with mobile cloud[J]. IEEE Transactions on Cloud Computing, 2016, 1–1.

[9] L. Yang, J. Cao, S. Tang, T. Li, and A. Chan. A framework for partitioning and execution of data stream applications in mobile cloud computing[J]. IEEE 5th International Conference on Cloud Computing (CLOUD), 2012, 794–802.