Noname manuscript No.
(will be inserted by the editor)

Partial Offloading Strategy for Mobile Edge Computing Considering Mixed Overhead of Time and Energy

Qiang Tang · Haimei Lyu · Guangjie Han · Jin Wang · Kezhi Wang

Received: date / Accepted: date

Abstract Mobile Edge Computing (MEC) utilizes wireless access network to provide powerful computing resources for mobile users to improve the user experience, which mainly includes two aspects: time and energy consumption. Time refers to the latency consumed to process user tasks, while energy consumption refers to the total energy consumed in processing tasks. In this paper, the time and energy consumption in user experience are weighted as a mixed overhead and then optimized jointly. We formulate a Mixed Overhead of Time and Energy (MOTE) minimization problem, which is a non-linear programming (NLP) problem. In order to solve this problem, the Block Coordinate Descent (BCD) method to deal with each variable step by step is adopted. We further analyze the minimum value of delay parameters in the model, and examine two special cases: 1-offloading and 0-offloading. In 1-offloading, all the task data is offloaded to MEC server, and no data offloaded in 0-offloading. The necessary and sufficient conditions for the existence of two special cases are also deduced. Besides, the multi-user situation is also discussed. In the performance evaluation, we compare MOTE with other offloading schemes, such as Exhaustive Strategy (ES) and Monte Carlo Simulation (MCS) method based strategy to evaluate the optimality. The simulation results show that MOTE always achieves the minimal overhead compared to other algorithms.

Keywords Full granularity · Partial offloading · Mixed overhead · Mobile edge computing

This work was supported in part by the National Key Research and Development Program, No.2017YFE0125300 and the National Natural Science Foundation of China-Guangdong Joint Fund under Grant No. U1801264, the Jiangsu Key Research and Development Program, No.BE2019648, in part by the Open fund of State Key Laboratory of Acoustics under Grant SKLA201901, in part by the National Natural Science Foundation of China (Grant Nos. 61772087, 61303043), in part by the Outstanding Youth Project of Hunan Province Education Department (Grant No. 18B162), and in part by the Double First-class International Cooperation and Development Scientific Research Project of Changsha University of Science and Technology (Grant No. 2018IC23).

Qiang Tang, Haimei Lyu, Jin Wang
Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation
School of Computer and Communication Engineering
Changsha University of Science and Technology, Changsha, 410114, Hunan, China

Guangjie Han
School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, 266061, China.
Department of Information and Communication Systems, Hohai University, Changzhou, 213022, China.
State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing, 100190, China.
Tel.: +86-15806128618
E-mail: hanguangjie@gmail.com
Guangjie Han is the corresponding author

Kezhi Wang
Department of Computer and Information Sciences
Northumbria University, Newcastle upon, UK

1 Introduction

In recent years, the fifth generation (5G) of mobile communication systems have emerged to cope the explosive growth of mobile data traffic, massive device connections and new services etc [1] [2]. Many typical application scenarios involves the 5G technology, such as the applications in dense residential areas, offices, stadiums, subways etc, and the various applications could be augmented reality, virtual reality, ultra-high definition video, cloud storage etc, which in general require for high speed and low latency. Faced with the

requirements, Mobile Cloud Computing (MCC) came into play, which refers to that user equipment accesses computing resources such as storage, computing, and database through remote connections. However, MCC is often far away from the end users, causing users to experience long delays. Therefore, to overcome this problem, MEC has been proposed as a promising solution [3] [4].

With the development of the Internet of Things (IoT), many devices have to exchange lots of information safely in real time, which requires very high performance of the network and communication environment. In the network layer, the wireless routing protocols are adopted to collect data [5], and in many scenarios they have different properties such as identifying and measuring jamming areas [6], protecting source node location [7], charging the sensor node [8], considering the energy constraint [9], and selecting the relay node on curve road [10] and intersection [11] for mobile wireless network.

Unlike the routing and data collection algorithms in the network layer of IoT, MEC is more relevant to underlying communication. Specifically, MEC refers to the computing capability sunk to a distributed base station, and conducts tasks such as calculation, storage, and processing for the IoT applications, such as video analyzing, indoor location [12], smart charging [13] [14] and demand response in smart grid [15]. Hence, the conventional wireless base station is upgraded to an computing competent base station [16]. The base station enables data to be processed on the wireless network side without being transmitted to a remote cloud server. Therefore, many researchers have paid attention to the task offloading of MEC. However, there are many issues to be considered in task offloading, such as service latency, energy consumption, caching management, computing resource distribution, the profit maximization of mobile service provider [17], and cooperative resource allocations [18] etc. For the mobile users, how to minimize the time or latency of the task execution and energy consumption of the user equipment is also an important issue.

2 Related Works

The computing offloading model in MEC can be divided into full offloading model and partial offloading model according to whether the tasks are separable or not.

2.1 Full Offloading Model

In this full offloading model, tasks are highly integrated, which means all the tasks can only be executed locally or offloaded to the MEC server as a whole. Full offloading model is also called binary offloading [3].

In [19], a theoretical framework of energy-optimal mobile cloud computing under stochastic wireless channel was provided. The goal was to save energy by performing mobile application tasks on mobile devices or offloaded to the edge cloud. It was divided into two sub-problems. The first one was solved at the user equipment by optimizing the CPU frequency. The server solves the second one by optimizing the transmission data rate. In [20], Wu et al. explored the tradeoff between shortening execution time and extending battery life of mobile devices, and based on which a novel adaptive full offloading scheme was proposed to select a optimal MEC server to offload all tasks. In [21], a novel solution that seamlessly integrates two technologies, mobile cloud computing and microwave power transfer (MPT), was proposed. In mobile cloud computing, the full offloading model was presented for mobile user to execute the task locally or offload the data to the MEC server. In MPT, the mobile user harvested the energy from base station to execute local computing. In [22], S. Barbarossa et al. analyzed different scenarios with the delay constraint, computation constraint to optimize the energy consumption. Both the single user and multi-users scenarios were analyzed. In [23], Wang et al. studied the joint energy minimization and resource allocation in C-RAN with MCC under the time constraints of the given tasks, and a multiple users full offloading model was formulated as a non-convex problem, which was transformed into convex problem and solved iteratively. Sun et al. in [24] formulated an optimization problem with the objective to maximize the sum of computation efficiency among users with weighted factors. The problem was solved efficiently with the iterative and gradient method, and the relationship between the size of data volume and local computation and data offloading was revealed.

2.2 Partial Offloading Model

One user may have served tasks to be processed, and the partial offloading refers to that the user's tasks can be partially offloaded to the MEC server and partially executed locally for better user experience.

Hao et al. in [25] proposed a hybrid linear programming problem with joint optimization of task caching and offloading, which was splited into two sub-problems and solved separately by using the branch and bound

method. Ren et al. in [26] studied a resource allocation problem to minimize the delay. Firstly, a closed-form expression for optimizing the data segmentation strategy was obtained. Based on this expression, the original problem was divided into a sub problem, which was solved by using the sub-gradient algorithm. Cao et al. [27] proposed a three-node mobile edge computing system including user nodes, assistant nodes and access nodes, and applied a four-slot protocol to jointly calculate and offload cooperation. Jia et al. [28] developed a heuristic program partitioning algorithm to minimize the execution latency by leveraging the overhead balancing concept between mobile users and servers, and proposed a polynomial-time approximate solution with guaranteed performance. In [29] Liu et al. proposed a Markov decision process approach to offload the tasks in the queue buffer to the MEC server or not. The average delay and power consumption of each task were analyzed, and a power-constrained delay minimization problem was formulated and solved. In [30], the authors introduced a wireless aware joint scheduling and computation offloading (JSCO) for multicomponent applications to shorten execution times by parallel processing appropriate components in the mobile and cloud. Mao et al. in [31] planned a stochastic optimization problem to achieve energy efficiency tradeoffloading in order to solve the randomness of channel conditions and task arrival. In [32], a comprehensive computation offloading solution using multiple radio links was proposed. An energy minimization problem for mobile devices was formulated and solved iteratively, and the local optimal solution was obtained. Zhang et al. in [33] researched the energy consumption minimization problems on mobile device and meeting a time deadline by strategically offloading tasks to the cloud. The optimization problem was formulated as a constrained shortest path problem, and approximately solved by the LARAC (Lagrangian Relaxation Based Aggregated Cost) algorithm.

2.3 Full Granularity Partial Offloading Model

The partial offloading models introduced above are mainly related to multiple tasks partial offloading. In practice, a single task can also be partially offloaded. For example, assume analyzing a video is a task. We can divide the video file into several segments and upload them to different servers to analyze, which can improve the efficiency and reduce the user's waiting time. In addition, we can also perform data segmentation for virus scanning, image compression and other tasks. Since task data can be arbitrarily divided into multiple fragments and proceeded either locally or offloaded to MEC server, we call this partial offloading as full granularity partial offloading [34].

Recently, a few studies have been done on full granularity partial offloading. In [34], Wang et al. investigated partial computation offloading by jointly optimizing the computational speed and transmission power of mobile device, and offloading ratio to minimize the energy consumption of user as well as the latency of application. However, this work did not consider the optimization of energy consumption and latency at the same time, in other words this work did not reflect user preferences in terms of time and energy consumption in the objective function. Besides, the work also did not discuss the situation of multiple users. In [35], O. Munoz et al analyzed the energy-latency tradeoff from the point of the mobile users, and then formulated an energy minimization problem to optimize the communication and computing resources. Kao et al. in [36] formulated an NP-hard problem to minimize the application latency while meeting prescribed resource utilization constraints. They also proposed a novel fully polynomial time approximation scheme Hermes to approximately solve this problem, and also an online algorithm was proposed to guarantee the bounded performance gap compared to the optimal strategy. Wang et al. in [37] exploited a multi-antenna non-orthogonal multiple access (NOMA) technique for multiuser computation offloading, and a weighted sum-energy consumption minimization problem at all users subject to their latency constraints was formulated. The partial offloading and binary offloading situations were considered and solved approximately in this work.

As the description above, in the full granularity partial offloading model, few papers considered the joint optimization of time and energy consumption for the mobile user. In this paper, we consider a full granularity partial offloading model, where the input data of the task is divided into local execution and MEC server execution in any proportion. We focus on jointly minimizing the user's overhead consisting of weighted execution time and consumed energy, and the weight is the user preferences in terms of time and energy consumption. By using the BCD method, we obtained closed-form solutions for all variables. We further analyze the time interval during which the user actually perform the full granularity partial offloading. In addition, we find the sufficient and necessary conditions for mobile user to choose 0-offloading and 1-offloading. As for the multi-user situation, the solution discussion is proposed. Finally, the simulation results illustrated the optimality and the superiority of MOTE compared with exhaustion algorithm, Monte Carlo algorithm and other optimization methods.

The rest of this paper is organized as follows. In section 3 and 4, the system model and optimization problem are formulated and solved respectively. In section 5, we further analyze the optimization model. The numerical results are introduced in section 6. In section 7, we summarize this paper.

3 System Model

Let's consider a typical IoT scenario, where a mobile user such as a policeman has a laptop to find a criminal from a video file with large size, as shown in Fig.1., which consumes a lot of computing resource and time. If partial video data can be transferred to MEC server, the processing time will be greatly reduced.

According to the scenario in Fig.1, we consider a mobile edge computing scenario that consists of one user device and one MEC server, where user device has computationally intensive task. Additionally, MEC server has computing and storage resources that deployed at the edge of the wireless access network. Therefore, user device can offload their task data to the MEC server through the wireless channel. We use a tuple $A = (T, F, D)$ to represent the user's task A, where T is the delay threshold, and F represents the number of CPU cycles required to complete task A, and D is the input of task data size.

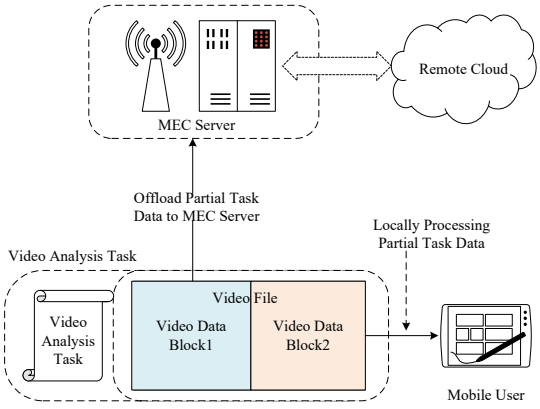


Fig. 1 A typical mobile edge computing scenario

3.1 Local Computing Model

For local computing tasks, we assume a linear relationship can be established between the number of CPU cycles F and the amount of data D bits input [34].

$$F = \alpha D \tag{1}$$

where $\alpha\,(\alpha > 0)$ depends on the nature of application, Since our partial offloading is a full-granular program partition. Thus, according to [35], we define $\lambda\,(0 \leq \lambda \leq 1)$ as the ratio, where λ is the ratio of the number of local execution data bits to the total number of input data bits. Therefore, the number of data bits executed locally is λD. The data size executed by the MEC server is $(1-\lambda)\,D$.

We define f_l as the user's CPU frequency, and f_{max} is the user's maximum CPU frequency. Then the constraint of f_l is:

$$f_l \leq f_{max} \tag{2}$$

Therefore, the user's local execution time T_l can be expressed as follows:

$$T_l = \frac{\alpha \lambda D}{f_l} \tag{3}$$

According to the reference [38], the energy consumption performed locally by the user can be expressed as:

$$E_l = k\alpha\lambda D f_l^2 \tag{4}$$

where k is the coefficient depending on the chip structure, we usually set $k = 10^{-26}$.

3.2 MEC Server Computing Model

According to the reference [39], there are two processes for the task execution in the MEC server computing: (1) task data is transmitted to the MEC server through the uplink channel. (2) task is executed in the MEC server. In the first phase, the user's uplink transmission rate r can be expressed as:

$$r = B\log_2(1 + \frac{ph_o}{\omega}) \tag{5}$$

where B is the bandwidth, ω is the noise power, h_o represents the channel gain from the user to the MEC server, p represents the transmission power, and p_{max} is the maximum transmission power. The constraint condition of p is:

$$p \leq p_{max} \tag{6}$$

Therefore, the transmission time T_u of the user task on the upstream channel is defined as follows:

$$T_u = \frac{(1-\lambda)\,D}{r} \tag{7}$$

In the second phase, we define f_c as the CPU frequency of the MEC server, and the execution time T_{ce} of the user in the MEC server is given by:

$$T_{ce} = \frac{\alpha\,(1-\lambda)\,D}{f_c} \tag{8}$$

Therefore, the total time T_c spent in the MEC server is the sum of time T_u and T_{ce} (Because the data size of the result is very small, the time taken to obtain the results is often neglected [40].):

$$T_c = \frac{(1-\lambda)\,D}{r} + \frac{\alpha\,(1-\lambda)\,D}{f_c} \tag{9}$$

The user's energy consumption in the MEC server can be expressed as follows:

$$E_c = pT_u = \frac{(1-\lambda)pD}{B\log_2(1+\frac{ph_o}{\omega})} \tag{10}$$

4 Problem Formulation

The objective function is defined from the perspective of mobile users. As for the mobile user, they only care the task execution delay and energy consumption. Although the execution delay and energy are two indicators in different dimensions, they can be weighted and then combined as a mixed indicator like other research work had done [38] [39]. We define overload U of a mobile user as a function consist of weighted time and energy consumption:

$$U = \beta\,(T_l + T_c) + (1-\beta)\,(E_l + E_c) \tag{11}$$

where $\beta\,(0 \le \beta \le 1)$ represents the weighting coefficient of the execution time for user task. The weighting factor of user energy consumption is $1-\beta$. The optimization problem is:

P1:

$$\begin{aligned} &\underset{f_l,p,\lambda}{\text{minimize}} \quad U \\ &s.t. \\ &C1:\ 0 \le \lambda \le 1 \\ &C2: 0 \le f_l \le f_{max} \\ &C3: 0 \le p \le p_{max} \\ &C4: T_l \le T \\ &C5: T_c \le T \end{aligned}$$

In **P1**, the constraint C1 represents the range of λ, and if $\lambda = 0$ all the task data is offloaded to the MEC server, while if $\lambda = 1$ all the task data is executed locally. The constraints C2 and C3 represent the maximum CPU frequency and the maximum transmission power of the mobile user. The constraint C4, C5 represent the local execution time constraint and the offloading time delay constraint respectively. Because the local execution and MEC server execution can be done in parallel and there is no correlation between the two parallel execution processes, then we limit the local execution time and MEC server execution time to C4 and C5 respectively.

One can see that **P1** is not a joint convex with regards to (w.r.t) f_l, p and λ, which is a nonlinear programming problem (NLP). In this paper, we firstly verify the convexity of NLP for each single variable, and then the Block Coordinate Descent (BCD) method is adopted to solve the NLP approximately.

4.1 Optimization of the local computing frequency f_l

In order to solve f_l, we get the following sub-problem:

P1.1:

$$\begin{aligned} &\underset{f_l}{\text{minimize}} \quad U_1 \\ &s.t. \\ &C2:\ 0 \le f_l \le f_{max} \\ &C4:\ T_l \le T \end{aligned}$$

where U_1 is:

$$\begin{aligned} U_1 &= \beta T_l + (1-\beta)E_l \\ &= \alpha\lambda D[\frac{\beta}{f_l} + (1-\beta)kf_l^2] \end{aligned} \tag{12}$$

Let:

$$y\,(f_l) = \frac{\beta}{f_l} + (1-\beta)kf_l^2 \tag{13}$$

Then we can get the second-order derivative of $y\,(f_l)$, which is $2\beta f_l^{-3} + 2(1-\beta)k > 0$. Besides, the constraints C2 and C4 are linear inequality, thus, **P1.1** is a convex problem w.r.t f_l. We obtain the optimal value of f_l:

$$f_l^* = \sqrt[3]{\frac{\beta}{2\,(1-\beta)\,k}} \tag{14}$$

According to the C2 and C4, we get the domain of f_l:

$$f_{min} \le f_l \le f_{max} \tag{15}$$

where $f_{min} = \frac{\alpha\lambda D}{T}$ deduced by C4. Finally, the bounded optimal value of f_l is:

$$f_l{}^* = \begin{cases} f_{min}, & if\ \sqrt[3]{\frac{\beta}{2\,(1-\beta)\,k}} < f_{min} \\ \sqrt[3]{\frac{\beta}{2\,(1-\beta)\,k}}, & if\ f_{min} \le \sqrt[3]{\frac{\beta}{2\,(1-\beta)\,k}} \le f_{max} \\ f_{max}, & if\ \sqrt[3]{\frac{\beta}{2\,(1-\beta)\,k}} > f_{max} \end{cases}$$

(16)

4.2 Optimization of the transmission power p

After obtaining the optimal solution ${f_l}^*$, we solve the variable p to get the optimal transmission power. The sub-problem is:

P1.2:

$$\begin{aligned} &\underset{p}{\text{minimize}} \quad U_2 \\ &s.t. \\ &C3: \ 0 \le p \le p_{max} \\ &C4: \ T_l \le T \end{aligned}$$

where U_2 is:

$$\begin{aligned} U_2 &= \beta T_u + (1-\beta) E_c \\ &= \frac{D(1-\lambda)(1-\beta)}{B} \frac{p + \frac{\beta}{1-\beta}}{\log_2\left(1 + \frac{p h_o}{\omega}\right)} \end{aligned} \tag{17}$$

In order to prove the U_2 is convex w.r.t p, we let:

$$f(p) = \frac{p + \frac{\beta}{1-\beta}}{\log_2\left(1 + \frac{p h_o}{\omega}\right)} \tag{18}$$

We further suppose $b = \frac{\beta}{1-\beta}$ and $a = \frac{h_o}{\omega}$. Then:

$$f(p) = \frac{p+b}{\log_2(1+ap)} \tag{19}$$

Lemma 1 *$f(p)$ is quasi-convex functions in the positive real number domain $\boldsymbol{R}^+$.*

Proof : According to reference [41], for quasi-convex functions defined on positive real number, the second derivative of the point where the first-order derivative is zero is non-negative.

Note that the first-order derivative of $f(p)$ is:

$$\frac{df(p)}{dp} = \frac{\log_2(1+ap) - \frac{a}{\ln 2} \cdot \frac{p+b}{1+ap}}{\log_2^2(1+ap)} \tag{20}$$

if we let the first derivative to zero, we get only one optimal point denoted by $\hat{p}$:

$$\hat{p} = \frac{ab-1}{lambertw\left(0, e^{-1}(ab-1)\right) \cdot a} - \frac{1}{a} \tag{21}$$

where $lambertw(\bullet)$ function is a product log function, and is a inverse function of $f(x) = x.e^x$. In this paper, $\hat{p}$ is always a positive real number. Besides, we also get the following condition by letting the first derivative of $f(p)$ as zero:

$$\log_2(1+a\hat{p}) = \frac{a}{\ln 2} \cdot \frac{\hat{p}+b}{1+a\hat{p}} \tag{22}$$

Then, we further get the second derivative of $f(p)$ as:

$$\begin{aligned} \frac{d^2 f(p)}{dp^2} = &\frac{a(2ap + ba\ln(ap+1) + 2ba)}{\ln^2(2)\left(\log_2(1+ap)\right)^3 (1+ap)^2} - \\ &\frac{a(ap\ln(ap+1) + 2\ln(ap+1))}{\ln^2(2)\left(\log_2(1+ap)\right)^3 (1+ap)^2} \end{aligned} \tag{23}$$

if we substitute condition (22) into (23), then we have:

$$\frac{d^2 f(\hat{p})}{d\hat{p}^2} = \frac{a^3 (b+\hat{p})^2}{(1+a\hat{p})^3 (\ln 2)^2 \log_2^3(1+a\hat{p})} \ge 0 \tag{24}$$

Therefore, this function is a quasi-convex function in the domain. ■

According to C3 and C5, we get the bounded domain of p:

$$p_{min} \le p \le p_{max} \tag{25}$$

where p_{min} is deduced by C5:

$$p_{min} = \left[2^{\frac{f_c(1-\lambda)D}{[Tf_c - \alpha(1-\lambda)D]B}} - 1 \right] \cdot \frac{\omega}{h_o} \tag{26}$$

Then, the optimal value of p is:

$$p^* = \begin{cases} p_{min}, & if\ p_{min} > \hat{p} \\ \hat{p}, & if\ p_{min} \le \hat{p} \le p_{max} \\ p_{max}, & if\ \hat{p} > p_{max} \end{cases} \tag{27}$$

4.3 Optimization of the offloading ratio λ

After optimizing the variables f_l and p, the rest variable is λ, the sub-problem w.r.t λ is:

P1.3:

$$\begin{aligned} &\underset{\lambda}{\text{minimize}} \quad U_3 \\ &s.t. \\ &C1: \ 0 \le \lambda \le 1 \\ &C4: \ T_l \le T \\ &C5: \ T_c \le T \end{aligned}$$

where U_3 is $\beta(T_l + T_c) + (1-\beta)(E_l + E_c)$. We can observed that **P1.3** is a linear program of λ, and we can get the upper bound and the lower bound according to C4 and C5 respectively:

$$1 - \frac{T}{D(\frac{1}{B\log_2(1+\frac{ph_o}{\omega})} + \frac{\alpha}{f_c})} \le \lambda \le \frac{Tf_l}{\alpha D} \tag{28}$$

Then according to the condition C1, lower bound of λ is:

$$\lambda_{min} = \max\left(0, 1 - \frac{T}{D(\frac{1}{B\log_2(1+\frac{ph_o}{\omega})} + \frac{\alpha}{f_c})}\right) \tag{29}$$

And the upper bound is:

$$\lambda_{max} = \min\left(1, \frac{Tf_l}{\alpha D}\right) \tag{30}$$

Then, we get the optimal value of λ:

$$\lambda^* = \begin{cases} \lambda_{max}, & if \ \ U_3(\lambda_{max}) \le U_3(\lambda_{min}) \\ \lambda_{min}, & if \ \ U_3(\lambda_{max}) \ge U_3(\lambda_{min}) \end{cases} \tag{31}$$

4.4 Overall Algorithm

When the variables f_l, p and λ are solved separately by using the BCD method, we can calculate out the optimal value of objective function in equation (11) by substituting the optimum values of this variables. Then the problem **P1** is solved approximately. The optimization progress is repeated some times until the objective function value is stable. MOTE algorithm is presented in **Algorithm 1**, where ε is a very small positive value.

5 Analysis

In this section, we firstly analyze the minimum delay for the parameter T. Then, we analyze the conditions for 0-offloading and 1-offloading.

Algorithm 1: Mixed Overhead of Time and Energy consumption (MOTE) Algorithm

Input: p^0, λ^0, ε
Output: f_l^*, p^*, λ^*
repeat
  Obtain the CPU frequency f_l^* according to (16);
  Obtain the transmission power p^* according to (27);
  Obtain the ratio λ^* according to (31);
  Calculate the value of $U^t(f_l^*, p^*, \lambda^*)$;
  $t = t + 1$;
until $\left|U^{(t)} - U^{(t-1)}\right| < \varepsilon$;
return f_l^*, p^*, λ^*.

5.1 Minimum delay threshold

In the sub-problem **P1.3**, the upper bound and lower bound of λ are determined by the parameter T. If we fixed other parameters and change T, we can get two curves for λ_{max} and λ_{min} respectively, which are shown in Fig.2:

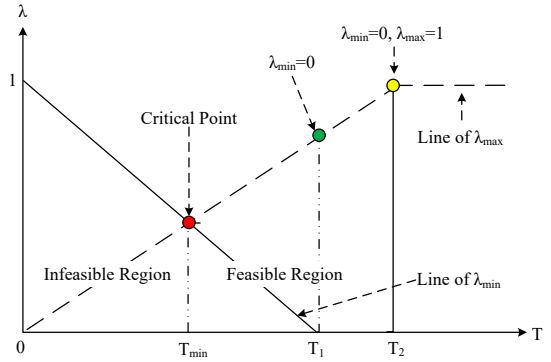


Fig. 2 Variation curves of feasible region with T

As shown in the figure above, the black broken line indicates λ_{max}, and the black solid line indicates λ_{min}. The intersection of these two lines is the minimum delay threshold T_{min}, which is:

$$T_{min} = \frac{1}{\frac{f_l}{\alpha D} + \frac{1}{(\frac{1}{B\log_2(1+\frac{ph_o}{\omega})} + \frac{\alpha}{f_c})D}} \tag{32}$$

Therefore, the delay threshold T we set must be greater than T_{min}. Otherwise this overhead optimization problem is not feasible. When delay threshold T is greater

than T_1, where T_1 is:

$$T_1 = D(\frac{1}{B\log_2(1+\frac{ph_o}{\omega})} + \frac{\alpha}{f_c}) \tag{33}$$

There may exist 0-offloading if λ selects the value of λ_{min}, while if λ selects the value of λ_{max} the full granularity partial offloading still exists.

When the delay threshold T is greater than T_2, where T_2 is:

$$T_2 = \frac{\alpha D}{f_l} \tag{34}$$

Then the offloading strategy is either 0-offloading or 1-offloading, and the optimal value of λ is either 1 or 0. Therefore if the partial offloading exists, i.e. the value of λ can be a decimal between 0 and 1, the delay threshold T should belong to the interval (T_{min},T_{max}), where the T_{max} is:

$$T_{max} = \min\left(\frac{\alpha D}{f_l}, D\left(\frac{1}{B\log_2(1+\frac{ph_o}{\omega})} + \frac{\alpha}{f_c}\right)\right) \tag{35}$$

In this subsection, we propose the conditions for the full granularity partial offloading and full offloading w.r.t parameter T. In the next subsection, we propose the sufficient and necessary conditions for 0-offloading.

5.2 Necessary and Sufficient Conditions for 0-offloading

In this section, we discuss the necessary and sufficient conditions for all task to be executed locally, which contains two conditions, which are (a) $\lambda = 1$ is a feasible solution, and (b) $\frac{dU(\lambda)}{d\lambda}\Big|_{\lambda=1} \leq 0$. According to condition (a) and equation (30), we have the following condition:

$$T \geq f_l \alpha D \tag{36}$$

In order to prove the validity of the condition (b) above, firstly we have to prove the function $U(\lambda)$ is convex, secondly we also have to prove the optimal function value can be obtained at the point $\lambda = 1$ shown in Fig.3.

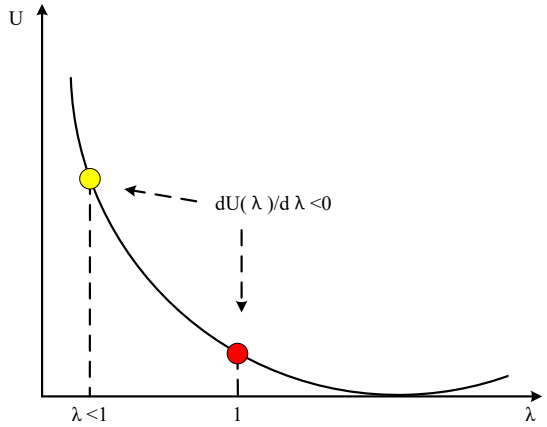


Fig. 3 The first derivative of function $U(\lambda)$ is less than 0

As shown in Fig.3, only when $\lambda = 1$, the optimal value of $U(\lambda)$ can be obtained. If we can further prove that $U(\lambda)$ is a convex function w.r.t λ, we can say the condition (a) and (b) are the necessary and sufficient conditions for local execution.

5.2.1 Convexity Proof

In order to prove the necessary and sufficient conditions, we should prove the convexity of the prime problem.

Theorem 1 *$U(\lambda, f_l, p)$ is a convex function w.r.t λ.*

Proof : Firstly, according to **P1.3** we can get the expression of $U(\lambda)$ as follows:

$$\begin{aligned} U(\lambda) = & \\ \beta & \left(\frac{\alpha\lambda D}{f_l^*} + \frac{(1-\lambda)D}{B\log_2\left(1+\frac{p^*h_o}{\omega}\right)} + \frac{\alpha(1-\lambda)D}{f_c}\right) + \\ (1-\beta) & \left(k\alpha\lambda D(f_l^*)^2 + \frac{(1-\lambda)p^*D}{B\log_2\left(1+\frac{p^*h_o}{\omega}\right)}\right) \end{aligned} \tag{37}$$

where p^* and f_l^* are given in equation (16) and (27) respectively. Although we can prove that $U(\lambda)$ is convex by the form of piecewise function, what we want to prove is that the original function $U(\lambda, f_l, p)$ is a convex function w.r.t λ. Besides, this function **P1.3** obtained is based on BCD method, which is the intermediate function of sub-optimal solution, and we need to prove that the prime problem is convex w.r.t λ.

The prime problem target function $U(\lambda, f_l, p)$ is:

$$U(\lambda, f_l, p) = \beta\left(\frac{\alpha\lambda D}{f_l} + \frac{(1-\lambda)D}{B\log_2\left(1+\frac{ph_o}{\omega}\right)} + \frac{\alpha(1-\lambda)D}{f_c}\right) + (1-\beta)\left(k\alpha\lambda D{f_l}^2 + \frac{(1-\lambda)pD}{B\log_2\left(1+\frac{ph_o}{\omega}\right)}\right) \tag{38}$$

which can be decomposed into two sub functions:

$$U_a(\lambda, f_l) = \beta\left(\frac{\alpha\lambda D}{f_l} + \frac{\alpha(1-\lambda)D}{f_c}\right) + (1-\beta)k\alpha\lambda D(f_l)^2 \tag{39}$$

and

$$U_b(\lambda, p) = \beta\frac{(1-\lambda)D}{B\log_2\left(1+\frac{ph_o}{\omega}\right)} + (1-\beta)\frac{(1-\lambda)pD}{B\log_2\left(1+\frac{ph_o}{\omega}\right)} \tag{40}$$

The $U_a(\lambda, f_l)$ is not joint convex w.r.t λ and f_l. We then use the variable substitution method and let:

$$x = \frac{\lambda}{f_l} \Rightarrow f_l = \frac{\lambda}{x} \tag{41}$$

Then (39) can be rewritten as:

$$U_a(\lambda, x) = \frac{\beta\alpha(1-\lambda)D}{f_c} + \alpha D\left(\beta x + (1-\beta)k\frac{\lambda^3}{x^2}\right) \tag{42}$$

The convexity of $U_a(\lambda, x)$ is determined by the following function:

$$g(\lambda, x) = \frac{\lambda^3}{x^2} \tag{43}$$

we can easily get its Hessian matrix and its characteristic values as 0 and $\frac{\lambda^2}{x^2}+1$, then the function $g(\lambda, x)$ is joint convex w.r.t λ and x. Therefore $U_a(\lambda, x)$ is a joint convex function. Because x has a convex set $x \in \left[0, \frac{1}{f_{min}}\right)$, then the $U_a(\lambda, x)$ is a convex function w.r.t λ according to the following lemma:

Lemma 2 *If f is convex in (x, y), and C is a convex nonempty set, then the function:*

$$g(x) = \inf_{y \in C} f(x, y) \tag{44}$$

is convex in x.

Proof : See [41] ■

Then we have proved that $U_a(\lambda, x)$ is a convex function w.r.t λ. In the following, we continue prove $U_b(\lambda, p)$ is also a convex function w.r.t λ.

We let:

$$x = \frac{1-\lambda}{\log_2\left(1+\frac{ph_o}{\omega}\right)} \Rightarrow p = \left(2^{\frac{1-\lambda}{x}} - 1\right)\frac{\omega}{h_o} \tag{45}$$

Then $U_b(\lambda, p)$ can be rewritten as:

$$U_b(x, \lambda) = \frac{D}{B}\left(\beta x + (1-\beta)2^{\frac{1-\lambda}{x}}x\frac{\omega}{h_o} - (1-\beta)x\frac{\omega}{h_o}\right) \tag{46}$$

The convexity of $U_b(x, \lambda)$ is determined by:

$$g(\lambda, x) = 2^{\frac{1-\lambda}{x}}x \tag{47}$$

we can easily get the characteristic values of its Hessian matrix, which are non-negative. So, $U_b(x, \lambda)$ is convex w.r.t x and λ. Because x belongs to a convex set, according to *Lemma 2*, the $U_b(\lambda, p)$ is a convex function w.r.t λ.

Besides the target function is convex, we can also easily find that the constraints C1, C2, C3, C4 and C5 are linear constraints w.r.t λ, x in $U_a(\lambda, x)$ and x in $U_b(x, \lambda)$, then the constraints are all convex.

In summary, $U(\lambda, f_l, p)$ is a convex function w.r.t λ, and the **Theorem 1** is proved. ■

5.2.2 Necessary and Sufficient Conditions Proof

According to the *Theorem 1*, we know the prime problem is convex w.r.t λ. If all the task data is executed locally, i.e $\lambda = 1$, then we can derive that (a) $\lambda = 1$ is a feasible solution, and (b) in the domain of λ, the target function $U(\lambda, f_l, p)$ are monotonically decreasing, i.e. $\frac{\partial U(\lambda, f_l, p)}{\partial \lambda} \leq 0$.

According to conditions (a) and (b), we can easily derive all the task data is executed locally.

5.3 Necessary and Sufficient Conditions for 1-offloading

Similar to the above subsection, if all the task data is executed at MEC server, two conditions can be derived: (c) $\lambda = 0$ is a feasible solution, and (d) in the domain of λ, the target function $U(\lambda, f_l, p)$ are monotonically increasing, i.e. $\frac{\partial U(\lambda, f_l, p)}{\partial \lambda} \geq 0$.

According to conditions (c) and (d), we can easily derive all the task data is executed at MEC server.

5.4 Discussion about Multi-User Situation

Except for a single user, multi-user scenarios are very common. As for this situation, our single user's model MOTE is still suitable.

In the multi-users scenario, we assume there are N mobile users, and we formulate the optimization problem as minimizing the sum of all the users' overhead:

P2:

$$\begin{aligned}
&\underset{f_{c,i},f_{l,i},p_i,\lambda_i}{\text{minimize}} \quad \sum_{i=1}^{N} U_i \\
&s.t. \\
&C6: \ 0 \le \lambda_i \le 1, \ 1 \le i \le N \\
&C7: 0 \le f_{l,i} \le f_{max,i}, \ 1 \le i \le N \\
&C8: 0 \le p_i \le p_{max,i}, \ 1 \le i \le N \\
&C9: T_{l,i} \le T_i, \ 1 \le i \le N \\
&C10: T_{c,i} \le T_i, \ 1 \le i \le N \\
&C11: 0 \le \sum_{i=1}^{N} f_{c,i} \le f_{c,max} \\
&C12: 0 \le f_{c,i} \le f_{c,max}, \ 1 \le i \le N
\end{aligned}$$

where U_i is $\beta_i \left(T_{l,i} + T_{c,i}\right) + (1-\beta_i)\left(E_{l,i} + E_{c,i}\right)$. λ_i is the offloading ratio for user i. $f_{l,i}$ is the local computing frequency of user i. $f_{c,i}$ is the MEC computing frequency for user i, and $f_{c,max}$ is the maximum computing frequency of MEC server. p_i is the transmission power for user i, and $p_{max,i}$ is the maximum transmission power for user i. $T_{l,i}$ and $T_{c,i}$ are the time of local computing and MEC computing respectively.

In the problem **P2**, all the mobile users are coupled by the constraint C11. According to equation (9) and (11), one can find that if the value of $f_{c,i}$ is maximized the overhead of user i is minimized. Then the equation is valid for the constraint C11. After combining the C11 and C12 together, we can obtain a new equation constraint: $f_{c,i} = f_{c,max} - \sum_{j=1}^{N} f_{c,j(j\neq i)}$.

At the beginning, we assume there is a MEC frequency distribution algorithm to distribute all the MEC computing frequency to each mobile user. Let's take user i as an example, and its initial MEC frequency is $f_{c,i}^0$. According to above analysis, the $f_{c,i}$ should be maximized, which means the optimal value of $f_{c,i}$ is $f_{c,i}^0$, and $f_{c,i}$ is solved. Then, the problem **P2** can be converted into:

P3:

$$\begin{aligned}
&\underset{f_{l,i},p_i,\lambda_i}{\text{minimize}} \quad \sum_{i=1}^{N} U_i \\
&s.t. \\
&C6: \ 0 \le \lambda_i \le 1, \ 1 \le i \le N \\
&C7: 0 \le f_{l,i} \le f_{max,i}, \ 1 \le i \le N \\
&C8: 0 \le p_i \le p_{max,i}, \ 1 \le i \le N \\
&C9: T_{l,i} \le T_i, \ 1 \le i \le N \\
&C10: T_{c,i} \le T_i, \ 1 \le i \le N
\end{aligned}$$

The problem **P3** is separable, then we have the following N sub-problems:

P3.1:

$$\begin{aligned}
&\underset{f_{l,i},p_i,\lambda_i}{\text{minimize}} \quad U_i \\
&s.t. \\
&C6: \ 0 \le \lambda_i \le 1 \\
&C7: 0 \le f_{l,i} \le f_{max,i} \\
&C8: 0 \le p_i \le p_{max,i} \\
&C9: T_{l,i} \le T_i \\
&C10: T_{c,i} \le T_i
\end{aligned}$$

where $1 \le i \le N$. The problem **P3.1** can be solved by using the BCD method proposed in **Algorithm 1** separately to get an approximate optimal value. Then the sum of all the users' objective function values is obtained for the primal problem **P2**.

6 Numerical Results

6.1 Benchmarks

We propose other two schemes, i.e. the one Without local Computing Frequency Optimization (WCFO), and the one Without the Transmission Power Optimization (WTPO). In WCFO, the computing frequency f_l is set as $0.5f_{max}$. In WTPO, the transmission power p is set as $0.5p_{max}$.

In order to evaluate the optimality of MOTE, an Exhaustive Strategy (ES) is proposed. In ES, each domain of variables f_l, p and λ is discretized into 1000 points, then the total number of partial offloading strategies is 10^{12}. All the strategies in ES will be calculated and the minimal overhead of all the strategies can be obtained for the optimal global minimal overhead of ES.

Besides, a Monte Carlo Simulation (MCS) method is adopted to search the optimal value of **P1**. We use uniform distribution to sample data from each variable's domain. The simulation times is set as 10^8, and for each simulation, we can update the minimal value of **P1** until the simulation is ended.

6.2 Environment Parameters Settings

We set $\alpha = 2.7$ to fit the computing features [35], the bandwidth B is set as 10^6 bps, channel gain h_o is set as -30 dB and noise power ω is set as -60 dBm. Besides, we also set the value of ε as 10^{-8}.

The simulation scenario parameters such as data size D, maximum transmission power p_{max}, et al. are set in each simulation subsection.

6.3 Optimality of MOTE

In order to evaluate the optimality of MOTE, we compare the objective function value of MOTE with that of ES and MCS. We set the data size D as 1.6×10^6 bits. p_{max} is 0.2 W, and $f_{l,max}$ is 2×10^7 Hz. The MEC server's CPU frequency f_c is 8×10^8 Hz. In order to make sure the feasibility of all the strategies in ES and MCS, we set a big delay threshold T, which is 6 seconds. Besides, the parameter β varies from 0.05 to 0.95 with the step as 0.10. The simulation results are presented in Fig.4. As we can see from the Fig.4, the objective

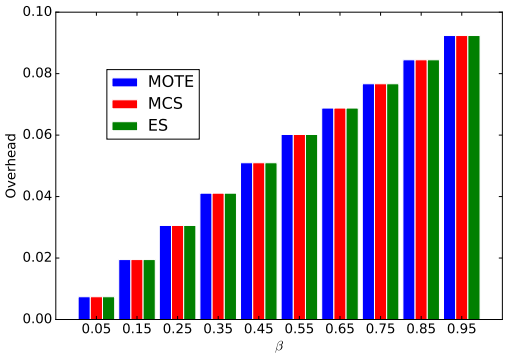


Fig. 4 The optimality of MOTE

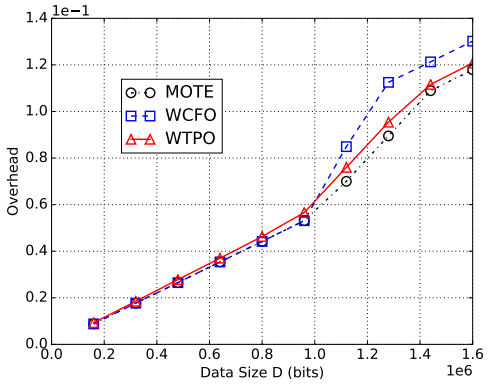
function value of MOTE is the same as that of ES and MCS. Although ES and MCS have not calculated all the possible offloading strategies, it's very likely that their objective function values are the same as the global optimal objective function values, which proves that the MOTE strategy is almost a global optimal strategy.

We can also find that the MOTE objective function value increases in a positive proportion with β. In fact, when the value of β is less than 0.01 or less, the MOTE objective function value will not increase in proportion to β, but will increase with the increase of β.

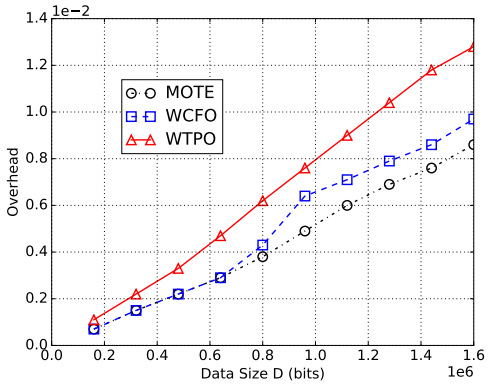
In the following subsections, we compare MOTE with WCFO and WTPO to illustrate the advantages of MOTE.

6.4 Changing Data Size D

In this subsection, the data size D varies from 1.6×10^5 bits to 1.6×10^6 bits. The delay threshold T is set as 0.06 seconds. The maximum transmission power p_{max} is 0.2 W. The user's local maximum CPU frequency $f_{l,max}$ is 2×10^7 Hz. The MEC server's CPU frequency f_c is 8×10^8 Hz. We simulate two circumstances where $\beta = 0.9$ and $\beta = 0.05$. The simulation results are shown in Fig.5.



(a) $\beta = 0.9$



(b) $\beta = 0.05$

Fig. 5 The relationship between overhead and data size.

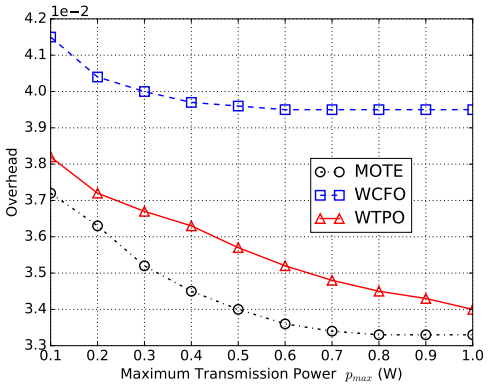
As we can see from the Fig.5(a), the overhead of MOTE is the best among these three offloading strategies. At the first six points, the MOTE offloads all the

task data to the MEC server, which is the same as strategy WCFO, thus the overheads of MOTE and WCFO are the same at the first six points. At the last four points, MOTE, WCFO and WTPO perform full granularity partial offloading, and our strategy MOTE performs best.

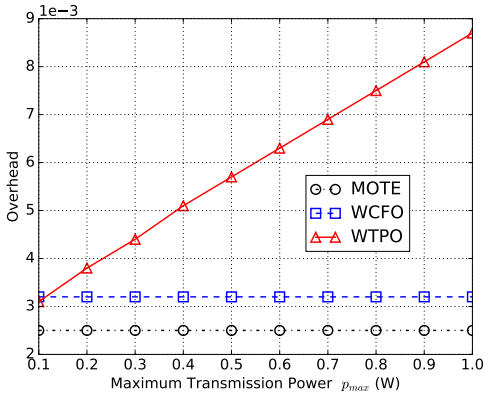
In Fig.5(b), we can see that the MOTE also performs the best among these strategies, although at some points the results of MOTE are overlap with that of WCFO.

6.5 Changing Maximum Transmission Power p_{max}

In this subsection, p_{max} varies from 0.1 W to 1.0 W with the step as 0.1 W. The data size D is 4.8×10^5 bits. f_{max} is 2×10^7 Hz. f_c is 8×10^8 Hz. We also simulate two scenarios where $\beta = 0.9, T = 0.02$ seconds and $\beta = 0.05, T = 0.03$ seconds. The simulation results are presented in Fig.6.



(a) $\beta = 0.9$



(b) $\beta = 0.05$

Fig. 6 The relationship between overhead and maximum transmission power.

As we can see from Fig.6, when $\beta = 0.9$, MOTE performs best among the three strategies. All the three strategies execute the full granularity partial offloading.

In the Fig.6(b), the $\beta = 0.05$, all the three strategies execute full granularity partial offloading, and our strategy MOTE is the best. Because the parameter p_{max} is only an upper bound, and can not affect the offloading ratio λ directly, then the MOTE and WCFO, which should optimize the transmission power p, always have the same transmission powers at the simulated 10 points. Thus the overheads of MOTE and WCFO at different points are the same. As for the WTPO, the offloading ratio λ is almost zero, and because the transmission power p is always $0.5p_{max}$, then the overhead line is approximately in proportion to p_{max}.

6.6 Changing Maximum Local Computing Frequency f_{max}

In this subsection, the f_{max} varies from 5×10^6 Hz to 5×10^7 Hz with the step as 5×10^6 Hz. The data size D is 4.8×10^5 bits. T is 0.03 seconds. f_c is 8×10^8 Hz. p_{max} is 0.2 W. We simulate the scenarios with the $\beta = 0.9$ and $\beta = 0.05$ respectively. Simulation results are shown in Fig.7.

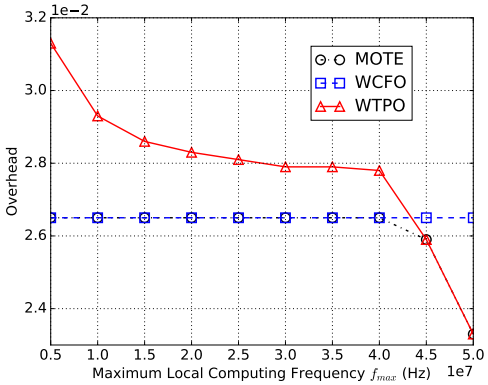
According to Fig.7(a), we find that MOTE always overlaps with WCFO or WTPO, and at all the 10 points, MOTE performs best. According to the values of λ at these 10 points, we find that the λ values in MOTE are 0 at the first 8 points and are 1 at the last 2 points, which means MOTE performs the full offloading in this scenario.

According to Fig.7(b), the overhead of MOTE is the minimum among the strategies. At the first 8 points, MOTE executes the full granularity partial offloading, and its overhead is the best. At the last 2 points, MOTE executes local computing, i.e. no data offloaded to the MEC server, and has the same overhead with that of WTPO.

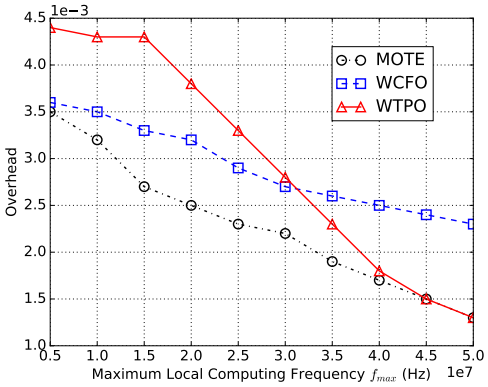
6.7 Changing MEC Server's Computing Frequency f_c

In this subsection, we change the f_c from 5×10^8 Hz to 50×10^8 Hz with the step as 5×10^8 Hz. The data size D is 4.8×10^5 bits. f_{max} is 2×10^7 Hz. p_{max} is 0.2 W. Two scenarios are simulated where the delay threshold T and β are set as 0.02 seconds and 0.9, 0.03 seconds and 0.05 respectively. The simulation results are presented in Fig.8.

In the Fig.8(a), all the three strategies execute full granularity partial offloading, the overhead of MOTE

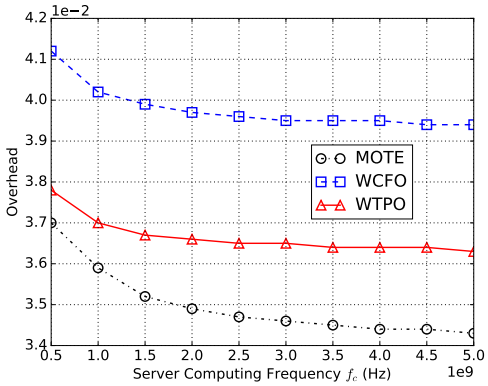


(a) $\beta = 0.9$

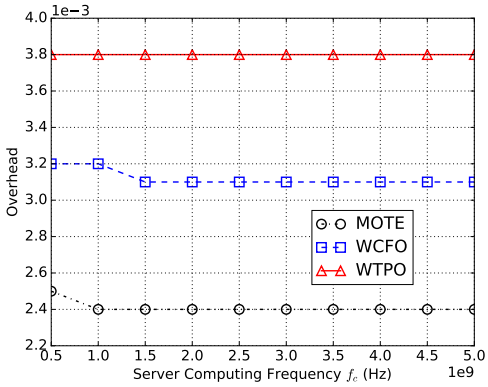


(b) $\beta = 0.05$

Fig. 7 The relationship between overhead and maximum local computing frequency.



(a) $\beta = 0.9$



(b) $\beta = 0.05$

Fig. 8 The relationship between overhead and MEC server's computing frequency.

is the minimum, and the same circumstances happens in the Fig.8(b).

Because the parameter f_c only affects the computing time at the MEC server, and its weight is β, then if β is big, such as 0.9, its increase will cause the overhead decrease, and the trend has been verified in Fig.8(a). Otherwise, if β is very small, such as 0.05, then its increase will not cause the overhead decrease obviously, shown in Fig.8(b).

6.8 Changing Delay Threshold T

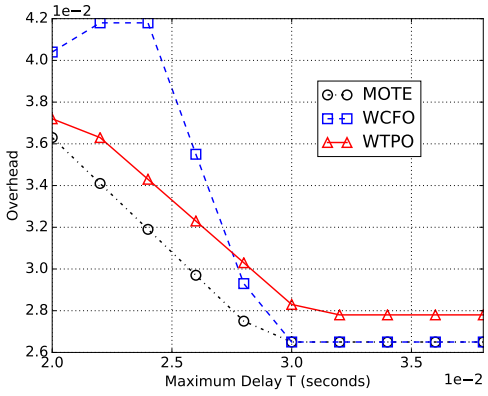
In order to evaluate the relationship between the delay threshold T and overhead, we change the value of T from 0.02 seconds to 0.038 seconds. The data size D is 4.8×10^5 bits. f_{max} is 2×10^7 Hz. f_c is 8×10^8 Hz. p_{max} is 0.2 W. The scenarios with $\beta = 0.9$ and $\beta = 0.05$ are simulated. Results are shown in Fig.9.

In the Fig.9(a), the overhead of MOTE is the minimum, and at the last 5 points all the three strategies offload all data to the MEC server and MOTE have the same performance with that of WCFO. In the Fig.9(b), the three strategies execute the full granularity partial offloading, and MOTE performs best.

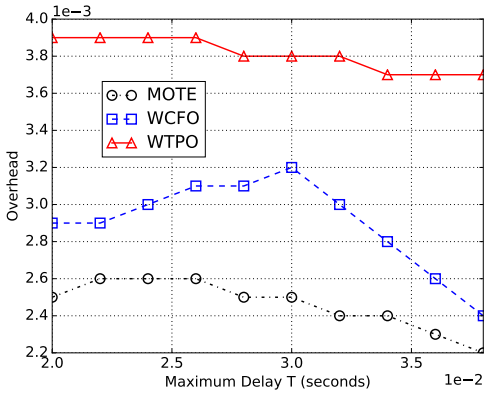
7 Conclusion

In this paper, we have proposed a mixed overhead full granularity partial offloading strategy for mobile user. The overhead combines the weighted time and energy consumption. The time includes the local computing time, data transmission time and computing time at MEC server. The energy consumption contains the local computing energy consumption and the data transmission energy. We have formulated a NLP problem and used the BCD method to solve it. We have analyzed the minimum delay threshold for T, and also analyzed the necessary and sufficient conditions for 0-offloading and 1-offloading. We have compared the performance of MOTE with that of ES and MCS to evaluate the optimality, and also compared with that of WCFO and WTPO to illustrate the advantages of MOTE.

We have discussed the multi-user scenario, and we find that the MOTE strategy can also be used in this scenario. But there is still lack of the initial MEC fre-



(a) $\beta = 0.9$



(b) $\beta = 0.05$

Fig. 9 The relationship between overhead and delay threshold.

quency distribution algorithm, which is adopted to distribute the optimal initial value of MEC computing frequency for each user and make the objective function minimized, which is a NP-hard problem and can be approximately solved by designing a heuristic algorithm. This work will be done in our future research.

Conflict of Interest: We declare that we have no conflict of interest.

References

1. Panwar, N., Sharma, S., and Singh, A. K., (2016). A survey on 5G: The next generation of mobile communication. Physical Communication, 18: 64-84.
2. Andrews, J. G., Buzzi, S., Choi, W., Hanly, S. V., Lozano, A., Soong, A. C., and Zhang, J. C., (2014). What will 5G be?. IEEE Journal on selected areas in communications, 32(6): 1065-1082.
3. Mao, Y., You, C., Zhang, J., Huang, K., and Letaief, K. B., (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys and Tutorials, 19(4): 2322-2358.
4. Mach, P., and Becvar, Z., (2017). Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys and Tutorials, 19(3): 1628-1656.
5. Han, G., Yang, X., Liu, L., and Zhang, W., (2018). A joint energy replenishment and data collection algorithm in wireless rechargeable sensor networks. IEEE Internet of Thing Journal, 5(4): 2596-2604.
6. Han, G., Liu, L., Zhang, W., and Chan, S., (2018). A hierarchical jammed-area mapping service for ubiquitous communication in smart communities. IEEE Communications Magazine, 56(1): 92-98.
7. Han, G., Wang, H., Jiang, J., Zhang, W., and Chan, S., (2018). CASLP: A confused arc-based source location privacy protection scheme in WSNs for IoT. IEEE Communications Magazine, 56(9): 42-47.
8. Han, G., Guan, H., Wu, J., Chan, S., Shu, L., and Zhang, W., (2018). An uneven cluster-based mobile charging algorithm for wireless rechargeable sensor networks. IEEE Systems Journal, doi: 10.1109/JSYST.2018.2879084.
9. He, S., Xie, K., Chen, W., Zhang, D., and Wen, J., (2018). Energy-Aware Routing for SWIPT in Multi-Hop Energy-Constrained Wireless Network. IEEE Access, 6: 17996-18008.
10. Cao, D., Liu, Y., Ma, X., Wang, J., Ji, B., Feng, C., Si, J., (2019). A Relay-Node Selection on Curve Road in Vehicular Networks. IEEE Access, 7: 12714-12728.
11. Cao, D., Zheng, B., Wang, J., Ji, B., and Feng, C., (2018). Design and analysis of a general relay-node selection mechanism on intersection in vehicular networks. Sensors, doi: 10.3390/s18124251.
12. Li, W., Chen, Z., Gao, X., Liu, W., and Wang, J., (2018). Multi-Model Framework for Indoor Localization under Mobile Edge Computing Environment. IEEE Internet of Things Journal, doi: 10.1109/JIOT.2018.2872133.
13. Tang, Q., Xie, M., Yang, K., Luo, Y., Zhou, D., Song, Y., (2018). A Decision Function based Smart Charging and Discharging Strategy for Electric Vehicle in Smart Grid. Mobile Networks and Applications, doi: 10.1007/s11036-018-1049-4.
14. Tang, Q., Wang, K., Luo, Y. and Yang, K., (2017). Congestion Balanced Green Charging Networks for Electric Vehicles in Smart Grid. In Proceedings of IEEE Global Communications Conference, 1-6.
15. Tang, Q., Yang, K., Zhou, D., Luo, Y., and Yu, F., (2016). A Real-time Dynamic Pricing Algorithm for Smart Grid with Unstable Energy Providers and Malicious Users. IEEE Internet of Things Journal, 3(4): 554-562.
16. Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., and Wang, W., (2017). A survey on mobile edge networks: Convergence of computing, caching and communications. IEEE Access, 5: 6757-6779.
17. Wang, X., Wang, K., Wu, S., Di, S., Jin, H., Yang, K., Ou, S., (2018). Dynamic Resource Scheduling in Mobile Edge Cloud with Cloud Radio Access Network. IEEE Transactions on Parallel and Distributed Systems, 29(11): 2429-2445.
18. Mei, H., Wang, K., and Yang, K., (2017). Multi-Layer Cloud-RAN With Cooperative Resource Allocations for Low-Latency Computing and Communication Services. IEEE Access, 5: 19023-19032.
19. Zhang, W., Wen, Y., Guan, K., Kilper, D., Luo, H., and Wu, D. O., (2013). Energy-optimal mobile cloud computing under stochastic wireless channel. IEEE Transactions on Wireless Communications, 12(9): 4569-4581.
20. Wu, H., Wang, Q., and Wolter, K., (2013). Tradeoff between performance improvement and energy saving in mobile cloud offloading systems. In Proceedings of IEEE Inter-

national Conference on Communications Workshops (ICC), 728-732.
21. You, C., Huang, K., and Chae, H., (2016). Energy efficient mobile cloud computing powered by wireless energy transfer. IEEE Journal on Selected Areas in Communications, 34(5): 1757-1771.
22. Barbarossa, S., Sardellitti, S., and Di Lorenzo, P., (2014). Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks. IEEE Signal Processing Magazine, 31(6): 45-55.
23. Wang, K., Yang, K., and Magurawalage, C. S., (2018). Joint energy minimization and resource allocation in C-RAN with mobile cloud. IEEE Transactions on Cloud Computing, 6(3): 760-770.
24. Sun, H., Zhou, F., and Hu, R. Q., (2019). Joint Offloading and Computation Energy Efficiency Maximization in a Mobile Edge Computing System. IEEE Transactions on Vehicular Technology, 68(3): 3052-3056.
25. Hao, Y., Chen, M., Hu, L., Hossain, M. S., and Ghoneim, A. (2018). Energy efficient task caching and offloading for mobile edge computing. IEEE Access, 6: 11365-11373.
26. Ren, J., Yu, G., Cai, Y., He, Y., and Qu, F., (2017). Partial offloading for latency minimization in mobile-edge computing. In proceedings of IEEE Global Communications Conference (GLOBECOM), 1-6.
27. Cao, X., Wang, F., Xu, J., Zhang, R., and Cui, S., (2018). Joint computation and communication cooperation for mobile edge computing. IEEE Internet of Things Journal, doi: 10.1109/JIOT.2018.2875246.
28. Jia, M., Cao, J., and Yang, L., (2014). Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing. In proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 352-357.
29. Liu, J., Mao, Y., Zhang, J., and Letaief, K. B., (2016). Delay-optimal computation task scheduling for mobile-edge computing systems. In proceedings of IEEE International Symposium on Information Theory (ISIT), 1451-1455.
30. Mahmoodi, S. E., Uma, R. N., and Subbalakshmi, K. P., (2016). Optimal joint scheduling and cloud offloading for mobile applications. IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2016.2560808.
31. Mao, S., Leng, S., Yang, K., Zhao, Q., and Liu, M., (2017). Energy efficiency and delay tradeoff in multi-user wireless powered mobile-edge computing systems. In proceedings of IEEE Global Communications Conference (GLOBECOM), 1-6.
32. Mahmoodi, S. E., Subbalakshmi, K. P., and Sagar, V., (2017). Cloud offloading for multi-radio enabled mobile devices. In Proceedings of IEEE International Conference on Communications Workshops (ICC), 5473-5478.
33. Zhang, W., Wen, Y., and Wu, D. O., (2015). Collaborative Task Execution in Mobile Cloud Computing Under a Stochastic Wireless Channel. IEEE Transactions on Wireless Communications, 14(1): 81-93.
34. Wang, Y., Sheng, M., Wang, X., Wang, L., and Li, J., (2016). Mobile-edge computing: Partial computation offloading using dynamic voltage scaling. IEEE Transactions on Communications, 64(10): 4268-4282.
35. Munoz, O., Pascual-Iserte, A., and Vidal, J., (2015). Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading. IEEE Transactions on Vehicular Technology, 64(10): 4738-4755.
36. Kao, Y., Krishnamachari, B., Ra, M., and Bai, F., (2017). Hermes: Latency Optimal Task Assignment for Resource-constrained Mobile Computing. IEEE Transactions on Mobile Computing, 16(11): 3056-3069.
37. Wang, F., Xu, J., and Ding, Z., (2019). Multi-Antenna NOMA for Computation Offloading in Multiuser Mobile Edge Computing Systems. IEEE Transactions on Communications, 67(3): 2450-2463.
38. Chen, X., (2015). Decentralized computation offloading game for mobile cloud computing. IEEE Transactions on Parallel and Distributed Systems, 26(4): 974-983.
39. Chen, X., Jiao, L., Li, W., and Fu, X., (2016). Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Transactions on Networking, 24(5): 2795-2808.
40. Cheng, Z., Li, P., Wang, J., and Guo, S., (2015). Just-in-time code offloading for wearable computing. IEEE Transactions on Emerging Topics in Computing, 2015, 3(1): 74-83.
41. Boyd, S., and Vandenberghe, L., (2004). Convex optimization. Cambridge university press. Cambridge.