Journal Pre-proof

Task number maximization offloading strategy seamlessly adapted to UAV scenario

Qiang Tang, Lu Chang, Kun Yang, Kezhi Wang, Jin Wang, Pradip Kumar Sharma

PII: S0140-3664(19)31387-8
DOI: https://doi.org/10.1016/j.comcom.2019.12.018
Reference: COMCOM 6075

To appear in: *Computer Communications*

Received date : 11 October 2019
Revised date : 1 December 2019
Accepted date : 8 December 2019

Please cite this article as: Q. Tang, L. Chang, K. Yang et al., Task number maximization offloading strategy seamlessly adapted to UAV scenario, *Computer Communications* (2019), doi: https://doi.org/10.1016/j.comcom.2019.12.018.

Task Number Maximization Offloading Strategy Seamlessly Adapted to UAV Scenario

Qiang Tang[1], Lu Chang[1], Kun Yang[2], Kezhi Wang[3], Jin Wang[1], and Pradip Kumar Sharma[4]
[1]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China
Email: tangqiang@csust.edu.cn, jinwang@csust.edu.cn
[2]School of Computer Science and Electronic Engineering, University of Essex, Essex, UK
Email: kunyang@essex.ac.uk
[3]Department of Computer and Information Sciences, Northumbria University, UK
Email: Kezhi.wang@northumbria.ac.uk
[4]Department of Computer Science and Engineering, Seoul National University of Science and Technology, South Korea
Email: pradip@seoultech.ac.kr
Jin Wang is the corresponding author

***Abstract*—Mobile edge computing (MEC) has been proposed in recent years to process resource-intensive and delay-sensitive applications at the edge of mobile networks, which can break the hardware limitations and resource constraints at user equipment (UE). In order to fully use the MEC server resource, how to maximize the number of offloaded tasks is meaningful especially for crowded place or disaster area. In this paper, an optimal partial offloading scheme POSMU (Partial Offloading Strategy Maximizing the User task number) is proposed to obtain the optimal offloading ratio, local computing frequency, transmission power and MEC server computing frequency for each UE. The problem is formulated as a mixed integer nonlinear programming problem (MINLP), which is NP-hard and challenging to solve. As such, we convert the problem into multiple nonlinear programming problems (NLPs) and propose an efficient algorithm to solve them by applying the block coordinate descent (BCD) as well as convex optimization techniques. Besides, we can seamlessly apply POSMU to UAV (Unmanned Aerial Vehicle) enabled MEC system by analyzing the 3D communication model. The optimality of POSMU is illustrated in numerical results, and POSMU can approximately maximize the number of offloaded tasks compared to other schemes.**

***Index Terms*—Offloaded Task Number Maximization, Partial Offloading, Unmanned Aerial Vehicle, Mobile Edge Computing.**

I. Introduction

MEC technology is gradually changing our world by offering powerful computing and communication resources to end users. In recent years, with the rapid development of information technologies (such as data gathering, coverage management and indoor localization technologies in wireless network [1] [2] [3], charging scheduling and security protection technologies in smart grid [4] [5] and routing technologies in vehicular network [6] etc), many applications need large amount of computing and communication resources to enhance their Quality of Experience (QoE). In order to address this problem, MEC technologies are considered as a feasible way.

UEs (such as mobile phones, smartphones, tablets, etc.) are terminal devices and can be used for learning, entertainment, reading, and working. However, users are often less satisfied with UEs due to their limited processing power, battery life and storage capacity [7]. With the emergence of cloud computing, many services (such as mobile healthcare, mobile learning, mobile gaming and mobile management) can be accessed from remote UEs [8], which improves the QoE but requires much computing and communication resources to reduce latency.

In order to further reduce latency and improve QoE, MEC is a feasible way by sinking resources to UEs side and processing the offloaded terminal tasks in real-time [9]. Task offloading includes decision, data uploading, MEC execution, and results return [10], which can effectively reduce the latency and energy consumption. According to the separability of tasks, computing offloading can be classified into binary offloading and partial offloading [11]. The binary offloading refers that tasks data cannot be further divided, and all the data should be offloaded to MEC server or executed locally, while the partial offloading means the task data can be partially proceed on MEC server, and other part of data is proceed locally.

In this paper, we study the partial offloading in the scenario with high UEs density. Each task has a latency limitation. The UEs energy as well as MEC resource are also limited. As the number of UEs increases, there are some UEs whose tasks cannot be offloaded and executed by MEC server. Then how to maximize the number of offloaded UEs' tasks is an important issue for this scenario. In this paper, we aim to address this problem by formulating it as a MINLP for single MEC server case. In order to make the MINLP traceable, it is converted and relaxed to multiple sub-problems. An overall solution POSMU is proposed by solving these sub-problems based on BCD method. Besides, we also analyze 3D communication model, and find that POSMU can be easily transplanted to UAV scenario.

II. Related Work

MEC has been studied extensively in recent years. According to the main objectives, the computing offloading related work has been summarized and untangled. Firstly, we introduce the work based on MEC system without UAV assistance, and then the work based on UAV enabled MEC system is introduced.

A. MEC System Without UAV Assistance

In MEC scenario, how to saving the UEs energy is an important issue for the energy sensitive UEs. In [12], by seamlessly integrating two technologies i.e. mobile cloud computing and microwave power transfer (MPT), You et al. proposed a novel framework for energy efficient computing, which aimed at minimizing the energy consumption for local computing and maximizing the energy savings for offloading. In their later work, an energy-efficient resource-management asynchronous mobile-edge computing offloading (MECO) system was studied in [13], where the mobile devices had heterogeneous input-data arrival time instants and computation deadlines. By introducing the concept of wireless aware joint scheduling and computing offloading (JSCO) for multi-component applications, in [14] Mahmoodi et al. made an optimal decision on which components need to be offloaded as well as the scheduling order of these components to save the energy of mobile devices. To minimize the total transmit energy of access point and subject to the constraints of computational tasks, in [15] Hu et al. studied a MEC system where two mobile devices were energized by the wireless power transfer (WPT) from an access point, and considered cooperative communications in the form of relaying via the nearer mobile device. By leveraging the variability in capabilities of mobile devices and user preferences, in [16] Guo et al. studied the energy efficient computing offloading management scheme in the MEC system, which minimized the energy consumption of all UEs via jointly optimizing computation offloading decision making, spectrum, power, and computation resource allocation. In [17] Mao et al. developed an online joint radio and computational resource management algorithm for multi-user MEC systems to minimize the long-term average weighted sum power consumption of the mobile devices and the MEC server. In [18] Dai et al. proposed a novel two-tier computing offloading framework in heterogeneous networks and formulated joint computing offloading and user association problem for multi-task mobile edge computing system to minimize overall energy consumption. In [19] Yang et al. formulated an energy optimization problem of offloading, which aimed to minimize the overall energy consumption at all system entities and took into account of the constraints from both computation capabilities and service delay requirement. In [20], a multi-subtasks-to-multi-servers model was proposed by Wang et al. to minimize all the energy consumption for subtasks. In [21] an offloading policy was proposed by Mazouzi et al. to decide which task should be offloaded and where to offloaded. The objective is minimize the total energy consumption for UEs.

In delay sensitive scenarios, the latency minimization is more important than energy saving and should be considered as the main objective. In [22] Ren et al. proposed a novel partial computing offloading model and then formulated a weighted-sum latency-minimization problem by optimally allocating the communication and computation resources, and in their follow-up work [23], a novel partial compression offloading system with joint communication and computation resource allocation was designed, which significantly reduced the end-to-end latency. With the development of caching technology, in [24] Yu et al. designed a collaborative offloading scheme and cached the popular computation results that was likely to be reused by other mobile users using caching enhancements to minimize the task latency at the mobile terminal side. In [25] Samanta et al. considered both the delay-tolerant and delay-constraint services in order to achieve the optimized service latency and revenue.

In some research scenarios, the energy consumption and latency are jointly minimized. In [26] Wang et al. investigated partial computing offloading by jointly optimizing the computational speed of smart mobile device (SMD), transmit power of SMD, and offloading ratio with two system design objectives: energy consumption of SMD minimization (ECM) and latency of application execution minimization (LM). In [27], Tang et al. proposed a partial offloading strategy MOTE considering the mixed overhead of time and energy, the time and energy are weighted and minimized.

B. UAV Enabled MEC System

In the UAV scenario, the 3D communication model is adopted. In general, UAV hovers in the air, and is far away from UEs to cover the ground UEs better. UAVs can be used as MEC servers to provide various computing offloading services for UEs. In [28], a novel framework of agent-enabled task offloading in UAV-aided MEC(UMEC) was put forth by Wang et al. The intelligence agent was guided to obtain the optimal offloading plan with aiming at minimum task execution delay and energy consumption. In [29], an UAV-enabled MEC system with wireless power transfer was studied by Du et al. In order to allow parallel transmission, a new time division multiple access (TDMA) based workflow model was proposed. The total energy consumption of UAV was minimized by optimizing the resource allocation, UAV hovering time etc. In [30], an UAV-enabled MEC network was proposed to provide services for UEs with fixed-wing and rotary-wing UAVs. The UAVs 3D trajectory and task cache strategies were optimized to minimize the energy consumption. In [31], an UAV-Enabled MEC system was studied by Zhou et al. to maximize the computation rate with the energy harvesting and UAV speed constraints. In [32] Yang et al. studied a MEC network with multiple UAVs. The user association, transmission power and computing capacity allocation, location planning were jointly optimized to minimize the sum power of UEs. In [33], a MEC system with UAV assistance was proposed by Hu et al. The weighted sum energy consumption of UAV and UEs were minimized based on considering the constraints of tasks, information-causality, bandwidth allocation and UAVs trajectory. According to the introduction above, a lot of research work is focus on energy consumption minimization problem, and the constraints are diverse, such as UAV trajectory, energy harvesting, UAV speed and so on.

C. Summary of Related Work and Our Contributions

According to the work presented above, the energy consumption minimization, latency minimization are the main research objectives both in the MEC or UAV-enabled MEC system. Few work had considered the offloaded task number maximization problem as the optimization target. In this paper, we will study an interesting problem with aiming at maximizing the offloaded task number in a crowded scenario. The problem is formulated as a MINLP, which can be solved approximately. The main contributions of this paper are summarized as follows.

- We consider a scenario with high density of UEs, and the MEC server is only one. In order to maximize the offloaded UEs' task number, a partial offloading optimization model is formulated.
- The optimization model is a MINLP, which is difficult to solve. By analyzing the property of this model, we decouple the primal problem into a series of sub-problems. These sub-problems can be solved separately by using the BCD method after analyzing the convexity for each variable.
- In order to apply our model to the scenario with UAV, we further analyze the 3D communication model, and illustrate that our model can be seamlessly transplanted to the UAV enabled MEC system.
- We evaluate the optimality for POSMU by comparing it with exhaustive algorithm and smart optimization algorithm respectively. We also compare POSMU with other schemes to illustrate its advantages.

The rest of this paper is organized as follows. System model is presented in section III. The problem is formulated and solved in section IV. In section V, we discuss the applicability of POSMU in UAV scenario, and numerical results are presented in section VI. Finally, the paper is summarized in section VII.

III. System Model

In this paper, we consider the scenario, where there are many UEs and few MEC servers. In our life, the crowded shopping malls and the disaster relief site have the characteristic. In this scenario, each UE want to upload its task to MEC server to get better experience. But, because of the limited MEC servers, many UEs computing offloading requirements cannot be met in a same time slot. Therefore, in order to maximize the utilization of MEC server in the crowded scenario, the offloaded task number maximization problem is studied in this work.

We assume that there are a set of N users $\mathcal{N} = \{1, 2, 3, ..., N\}$ and a mobile edge computing server (MEC server) in the scenario. For a UE task, the data size is D, and CPU cycles required to complete the task is F. UE also has a latency limitation τ for its task. As illustrated in Fig.1, which shows a typical MEC scenario consists of user devices and edge server, where UE can offload their intensive task to the MEC server. As shown in the figure, only one MEC server processes a lot of UEs tasks. Because of limited MEC resource, there must be some UEs tasks cannot be offloaded to MEC server. The MEC server is working at full capacity. Fig.2 shows our computing offloading model, which is a full granularity partial offloading, that is, for a certain task, only part of its data can be offloaded to the MEC server, and other data is locally proceeded.

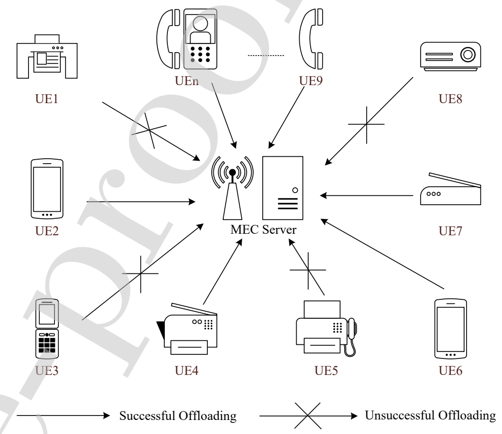

Fig. 1. The offloading scenario with massive UEs

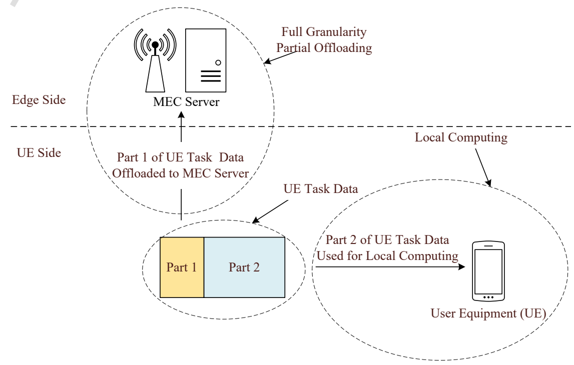

Fig. 2. The principle of full granularity partial offloading

A. Local Computing

As for UE i, the task data size is D_i, which consumes F_i CPU cycles to complete the task. In this paper, we assume there is a linear relationship between F_i and D_i [34]:

$$F_i = \alpha D_i \tag{1}$$

where α depends on the nature of application. As for the full granularity partial offloading [35], we define λ_i as a ratio for the non-offloaded data. Therefore, the number of data bits executed locally is $\lambda_i D_i$, and the data size executed by the MEC server is $(1 - \lambda_i)D_i$.

We let f_i^l denote the local computing frequency of UE i. Then the local task data processing time is:

$$T_i^l = \frac{\lambda_i F_i}{f_i^l} \tag{2}$$

According to [36]- [37], the energy consumption of UE i performed locally is:

$$E_i^l = k \cdot (f_i^l)^2 \lambda_i \cdot F_i \tag{3}$$

where k is the coefficient depending on the chip structure, and typically $k = 10^{-26}$.

B. MEC Server Computing

When UE offloads task on MEC server, the uplink data rate of UE i is:

$$r_i = B\log_2\left(1 + \frac{p_i h_i}{N_0}\right) \tag{4}$$

where p_i is the transmission power of UE i, and B represents the channel bandwidth. N_0 is the noise power. In this model, we assume there are many orthogonal sub-channels. Besides, the frequency reuse techniques or interference coordination are adopted to mitigate the interference [38]. So, the transmitting interference is not considered in this work. h_i is the channel gain between UE i and the MEC server. Because the MEC server is nearby the UEs, the distance influence is not considered in this model, and we assume all the UEs have the same channel gain h_i. Then, the task data transmission time is given as:

$$T_i^{tra} = \frac{(1-\lambda_i)\, D_i}{B\log_2(1 + \frac{p_i h_i}{N_0})} \tag{5}$$

As for the MEC server, we define f_i^c as the computing frequency of MEC server allocated to the task of UE i. Therefore the task data processing time on MEC server is:

$$T_i^{pro} = \frac{(1-\lambda_i)\, F_i}{f_i^c} \tag{6}$$

Then, the task time consumed on MEC server is:

$$\begin{aligned} T_i^c &= T_i^{tra} + T_i^{pro} \\ &= \frac{(1-\lambda_i)\, D_i}{B\log_2\left(1 + \frac{p_i h_i}{N_0}\right)} + \frac{(1-\lambda_i)\, F_i}{f_i^c} \end{aligned} \tag{7}$$

In this paper, we neglect the time delay for MEC server sending the processed results back to the UE. This is because the size of task result is much smaller than that of task data [39]. The UE's energy consumption in the offloading process is only the transmission energy:

$$E_i^{tra} = p_i \frac{(1-\lambda_i)\, D_i}{B\log_2(1 + \frac{p_i h_i}{N_0})} \tag{8}$$

C. Problem Formulation

In the optimization model, we consider how to maximize the offloaded task number of UEs, and assume that in one time slot one UE only has a task. The optimization problem is formulated as follows:

P1:

$$\underset{N, f_i^l, f_i^c, p_i, \lambda_i}{\text{maximize}} \quad N$$

s.t.

$$\begin{aligned}
&C1: \sum_{i=1}^{N} f_i^c \le f_{c,\max}, \forall i \in \mathcal{N} \\
&C2: \frac{\lambda_i F_i}{f_i^l} \le \tau_i, \forall i \in \mathcal{N} \\
&C3: k(f_i^l)^2 \lambda_i F_i + p_i \frac{(1-\lambda_i)\, D_i}{B\log_2\left(1 + \frac{p_i h_i}{N_0}\right)} \le E_i, \forall i \in \mathcal{N} \\
&C4: 0 \le p_i \le p_{i,max}, \forall i \in \mathcal{N} \\
&C5: 0 \le \lambda_i \le 1, \forall i \in \mathcal{N} \\
&C6: \frac{(1-\lambda_i)\, D_i}{B\log_2\left(1 + \frac{p_i h_i}{N_0}\right)} + \frac{(1-\lambda_i)\, F_i}{f_i^c} \le \tau_i, \forall i \in \mathcal{N} \\
&C7: \ 0 \le f_i^l \le f_{i,\max}^l, \forall i \in \mathcal{N}
\end{aligned}$$

where the constraint C1 ensures that the total required computing frequency for the offloaded tasks should not exceed the computing frequency capacity $f_{c,max}$ of MEC server. The constraint C2 represents that the local execution time should not bigger than the latency limitation τ_i of UE i. Constraint C3 means the total energy consumption should not exceed the total residual energy E_i of UE i. C3 contains two parts, which are local computing energy consumption and data transmission energy consumption. Constraints C4, C5 and C7 are the domains for p_i, λ_i and f_i^l respectively. $p_{i,max}$ is upper bound for p_i. $f_{i,\max}^l$ is upper bound for f_i^l. Constraint C6 means the time consumed of the offloaded task data should not exceed the latency limitation τ_i. C6 contains two parts, which are task data uploading time and executed time on the MEC server.

Both in C3 and C6, the transmission rate hasn't included the interference among UEs. The local computing and MEC server computing can be executed at the same time. C3 is the time constraint for local computing, and C6 is the time constraint for MEC computing. In this model, the energy consumption saving is not our main purpose, and energy consumption is only considered in C3. This is because in the crowded scenario with limited MEC resources how to maximize the offloaded UEs tasks number is meaningful. Besides, in order to control the energy consumption for a task, the value of E_i can be preset or limited by UEs.

IV. Problem Solving

According to **P1**, only C1 has coupled all the UEs variables f_i^c, and the constraints from C2 to C6 are all about single UE. Because the MEC server resource is limited in high density UEs scenario, and if the f_i^c is minimized while other constraints are satisfied, the offloaded UEs task number can be maximized. Then the primal problem **P1** can be converted into the following N sub-problems, which minimize the MEC server computing frequency f_i^c for each UE's task.

P2:

$$\begin{aligned}
&\underset{f_i^l, f_i^c, p_i, \lambda_i}{\text{minimize}} \ f_i^c \qquad \forall i \in \mathcal{N} \\
&\text{s.t.} \\
&\text{C2}: \frac{\lambda_i F_i}{f_i^l} \leq \tau_i \\
&\text{C3}: k(f_i^l)^2 \lambda_i F_i + p_i \frac{(1-\lambda_i) D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)} \leq E_i \\
&\text{C4}: 0 \leq p_i \leq p_{i,max} \\
&\text{C5}: 0 \leq \lambda_i \leq 1 \\
&\text{C6}: \frac{(1-\lambda_i) D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)} + \frac{(1-\lambda_i) F_i}{f_i^c} \leq \tau_i \\
&\text{C7}: \ 0 \leq f_i^l \leq f_{i,\max}^l
\end{aligned}$$

According to **P2**, if all the sub-problems are solved, the minimal MEC server computing frequency for UEs can be obtained, and then the maximal UE number N can be calculated by sorting ascending and summing all the f_i^cs without exceeding the maximal computing frequency of MEC server $f_{c,max}$. In the following sections, we only consider one sub-problem of UE i for simplicity.

Although **P2** is a NLP, the objective function only contains the variable f_i^c. According to C6, the variable f_i^c only exists in this condition, and a lower bound for f_i^c is:

$$f_i^c \geq \frac{(1-\lambda_i) F_i}{\tau_i - \dfrac{(1-\lambda_i) D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)}} \tag{9}$$

which needs an addition constraint C8 to make (9) hold:

$$\frac{(1-\lambda_i) D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)} - \tau_i \leq 0 \tag{10}$$

According to (9), the optimal solution for f_i^c can be obtained at the lower bounder. Then, the variable f_i^c is eliminated. The remaining problem is:

P3:

$$\begin{aligned}
&\underset{f_i^l, p_i, \lambda_i}{\text{minimize}} \ \frac{(1-\lambda_i) F_i}{\tau_i - \dfrac{(1-\lambda_i) D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)}} \\
&\text{s.t.} \\
&\text{C2}: \frac{\lambda_i F_i}{f_i^l} \leq \tau_i \\
&\text{C3}: k(f_i^l)^2 \lambda_i F_i + p_i \frac{(1-\lambda_i) D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)} \leq E_i \\
&\text{C4}: 0 \leq p_i \leq p_{i,max} \\
&\text{C5}: 0 \leq \lambda_i \leq 1 \\
&\text{C7}: \ 0 \leq f_i^l \leq f_{i,\max}^l \\
&\text{C8}: \ \frac{(1-\lambda_i) D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)} - \tau_i \leq 0
\end{aligned}$$

P3 is not a jointly convex problem with regard to (w.r.t) f_i^l, p_i and λ_i. But the problem is convex for a single variable, then we can use the block coordinated descent (BCD) method [40] to solve each variable individually to obtain an approximately optimal solution.

A. *Optimal Local Computing Frequency*

Because the variable f_i^l is not exist in the objective function, which results that the value of f_i^l can not effect the objective function directly. But it can effect the objective function by affecting other variables such as λ_i.

According to **P3**, C2 is a linear constraint for f_i^l, and C3 is a quadratic function constraint for f_i^l. The objective function do not contains f_i^l. Thus, **P3** is a convex problem with regard to f_i^l. The sub-problem is:

P3.1:

$$\begin{aligned}
&\underset{f_i^l}{\text{minimize}} \ \frac{(1-\lambda_i) F_i}{\tau_i - \dfrac{(1-\lambda_i) D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)}} \\
&\text{s.t.} \\
&\text{C2}: \frac{\lambda_i F_i}{f_i^l} \leq \tau_i \\
&\text{C3}: k(f_i^l)^2 \lambda_i F_i + p_i \frac{(1-\lambda_i) D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)} \leq E_i \\
&\text{C7}: \ 0 \leq f_i^l \leq f_{i,\max}^l
\end{aligned}$$

According to C2, C3 and C7, we get a new domain for f_i^l:

$$\frac{\lambda_i F_i}{\tau_i} \leq f_i^l \leq \bar{f}_i^l \tag{11}$$

where:

$$\bar{f}_i^l = \min\left(\sqrt{\frac{E_i - p_i \frac{(1-\lambda_i)D_i}{B\log_2\left(1+\frac{p_i h_i}{N_0}\right)}}{k\lambda_i F_i}}, f_{i,\max}^l\right) \tag{12}$$

According to **P3.1**, each UE should require the minimal MEC server's computing frequency to maximize the number of offloaded UEs tasks. Then f_i^l should be the upper bound to process as much task data as possible. If f_i^l is bigger, the less time it takes for local computing, which results the more task data proceeded locally and the smaller MEC server's computing resources requirement. Therefore, optimal solution of f_i^l is:

$$f_i^{l*} = \begin{cases} \bar{f}_i^l, \ if \ \bar{f}_i^l \geq \frac{\lambda_i F_i}{\tau_i} \\ no \ feasible \ solution, \ else \end{cases} \tag{13}$$

B. Optimal Local Computing Ratio

The local computing ratio λ_i is the ratio of locally proceeded data volume to total task data volume. According to **P3**, the objective function is a monotonically decreasing function w.r.t λ_i. The sub-problem is:

P3.2:

$$\underset{\lambda_i}{\text{minimize}} \ \frac{(1-\lambda_i)F_i}{\tau_i - \frac{(1-\lambda_i)D_i}{B\log_2\left(1+\frac{p_i h_i}{N_0}\right)}}$$

s.t.

$$\text{C2}: \frac{\lambda_i F_i}{f_i^l} \leq \tau_i$$

$$\text{C3}: k(f_i^l)^2\lambda_i F_i + p_i \frac{(1-\lambda_i)D_i}{B\log_2\left(1+\frac{p_i h_i}{N_0}\right)} \leq E_i$$

$$\text{C5}: 0 \leq \lambda_i \leq 1$$

$$\text{C8}: \frac{(1-\lambda_i)D_i}{B\log_2\left(1+\frac{p_i h_i}{N_0}\right)} - \tau_i \leq 0$$

In order to get the upper bound of λ_i, we calculate out a new value range according to C2, C3, C5 and C8.

According to C2, we have:

$$\lambda_i \leq \frac{\tau_i f_i^l}{F_i} \tag{14}$$

According to C3, we derive that:

$$\lambda_i q_i \leq h_i \tag{15}$$

where:

$$h_i = E_i - p_i \frac{D_i}{B\log_2\left(1+\frac{p_i h_i}{N_0}\right)} \tag{16}$$

$$q_i = \left(k(f_i^l)^2 F_i - p_i \frac{D_i}{B\log_2\left(1+\frac{p_i h_i}{N_0}\right)}\right) \tag{17}$$

Then, if $h_i \geq 0$ and $q_i \geq 0$, we have:

$$\lambda_i \leq \frac{h_i}{q_i} \tag{18}$$

if $h_i \geq 0$ and $q_i < 0$, we have:

$$\lambda_i \geq \frac{h_i}{q_i} \tag{19}$$

According to the value interval C5, we see that (19) is always hold and contained by C5. Then, the condition for (19) can be ignored.

if $h_i < 0$ and $q_i \geq 0$, we have:

$$\lambda_i \leq \frac{h_i}{q_i} \tag{20}$$

According to C5, we know that λ_i is a non-negative value, which contraries to (20). Then, (20) is not hold.

if $h_i < 0$ and $q_i < 0$, we have:

$$\lambda_i \geq \frac{h_i}{q_i} \tag{21}$$

According to C8, we have:

$$\lambda_i \geq 1 - \frac{\tau_i B\log_2\left(1+\frac{p_i h_i}{N_0}\right)}{D_i} \tag{22}$$

Therefore, we get the new upper and lower bounds for λ_i:

$$\lambda_{i,\max} = \begin{cases} \min\left\{\frac{\tau_i f_i^l}{F_i}, 1, \frac{h_i}{q_i}\right\}, \ if \ h_i \geq 0, \ q_i \geq 0 \\ no \ feasible \ value, \ if \ h_i < 0, \ q_i \geq 0 \\ \min\left\{\frac{\tau_i f_i^l}{F_i}, 1\right\}, \ otherwise \end{cases} \tag{23}$$

$$\lambda_{i,\min} = \begin{cases} \max\left\{0, s_i, \frac{h_i}{q_i}\right\}, \ if \ h_i < 0, \ q_i < 0 \\ \max\left\{0, s_i\right\}, \ otherwise \end{cases} \tag{24}$$

where:

$$s_i = 1 - \frac{\tau_i B\log_2\left(1+\frac{p_i h_i}{N_0}\right)}{D_i} \tag{25}$$

Then, the optimal value for λ_i is:

$$\lambda_i^* = \begin{cases} \lambda_{i,\max}, \ if \ \lambda_{i,\max} \geq \lambda_{i,\min} \\ no \ feasible \ solution, \ if \ \lambda_{i,\max} < \lambda_{i,\min} \end{cases} \tag{26}$$

C. Optimal Transmission Power

According to P3, the sub-problem for solving the transmission power is:

P3.3:

$$\underset{p_i}{\text{minimize}} \quad \frac{(1-\lambda_i)\,F_i}{\tau_i - \dfrac{(1-\lambda_i)\,D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)}}$$

s.t.

$$\text{C3}: k(f_i^l)^2\lambda_i F_i + p_i\frac{(1-\lambda_i)\,D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)} \le E_i$$

$$\text{C4}: 0 \le p_i \le p_{i,max}$$

$$\text{C8}: \frac{(1-\lambda_i)\,D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)} - \tau_i \le 0$$

In **P3.3**, the objective function is monotonically decreasing w.r.t p_i. Then, if we get a new value range for p_i according to C3, C4 and C8, the optimal value can be obtained.

According to C8, we have:

$$p_i \ge \frac{N_0\left(2^{\frac{(1-\lambda_i)\,D_i}{B\tau_i}}-1\right)}{h_i} \tag{27}$$

According to C3, we get the following condition:

$$g(p_i) = p_i\frac{(1-\lambda_i)\,D_i}{B\log_2\left(1+\dfrac{p_i h_i}{N_0}\right)} + k(f_i^l)^2\lambda_i F_i - E_i \le 0 \tag{28}$$

Theorem 1: $g(p_i)$ is a quasi-convex function in the domain.

Proof 1: According to [41], we firstly calculate the first derivative of $g(p_i)$:

$$g'(p_i) = \frac{(1-\lambda_i)\,D_i\left(\ln\left(1+p_i\dfrac{h_i}{N_0}\right)\left(1+p_i\dfrac{h_i}{N_0}\right)-p_i\dfrac{h_i}{N_0}\right)}{\ln(2)\,B\log_2\left(1+p_i\dfrac{h_i}{N_0}\right)^2\left(1+p_i\dfrac{h_i}{N_0}\right)} \tag{29}$$

Then, we let $g'(p_i)=0$, and get the following condition:

$$\ln\left(1+p_i\frac{h_i}{N_0}\right) = \frac{p_i\dfrac{h_i}{N_0}}{\left(1+p_i\dfrac{h_i}{N_0}\right)} \tag{30}$$

Then, we continue to calculate the second-order derivative of $g(p_i)$:

$$g''(p_i) = \frac{(1-\lambda_i)\,D_i\dfrac{h_i}{N_0}v}{\ln^2(2)\,B\log_2\left(1+p_i\dfrac{h_i}{N_0}\right)^3\left(1+p_i\dfrac{h_i}{N_0}\right)^2} \tag{31}$$

where v is:

$$v(p_i) = -p_i\frac{h_i}{N_0}\ln\left(1+p_i\frac{h_i}{N_0}\right)+2p_i\frac{h_i}{N_0}-2\ln\left(1+p_i\frac{h_i}{N_0}\right) \tag{32}$$

Then, substitute condition (30) into (32), and we have:

$$v(p_i) = \frac{1}{\left(1+p_i\dfrac{h_i}{N_0}\right)}\left(p_i\frac{h_i}{N_0}\right)^2 \ge 0 \tag{33}$$

Then the second-order derivative $g''(p_i)\ge 0$ when p_i is the solution for equation $g'(p_i)=0$. Thus, $g(p_i)$ is a quasi-convex function in the domain [41]. ■

In order to solve (28) to get a new bound for p_i, we convert the inequality (28) into:

$$g_1(p_i) = p_i(1-\lambda_i)\,D_i + B\log_2\left(1+\frac{p_i h_i}{N_0}\right)\left(k(f_i^l)^2\lambda_i F - E_i\right) \le 0 \tag{34}$$

We can see that when $p_i=0$, the $g_1(p_i)=0$, then $p_i=0$ is a lower bound of feasible solution for the (34). Because $g(p_i)$ is a quasi-convex function in the domain, there is an upper bound of feasible solution for (28), which is solved by using the binary search method presented in **Algorithm 1**.

Algorithm 1: Upper Bound Search Algorithm (UBSA)

Input: $p_{i,\max}, p_{i,\min}, k, f_i^l, \lambda_i, F_i, D_i, B, h_i, N_0, E_i, \varepsilon$
Output: $\bar{p}_i$

$p_0 = p_{i,\max}$;
calculate
$g(p_0) \leftarrow p_0\dfrac{(1-\lambda_i)\,D_i}{B\log_2\left(1+\dfrac{p_0 h_i}{N_0}\right)} + k(f_i^l)^2\lambda_i F_i - E_i$
if $g(p_0)\le 0$ **then**
  $\bar{p}_i = p_0$
end
if $g(p_0) > 0$ **then**
  $p_s = p_{i,\min}$;
  $p_t = p_0$;
  repeat
    $p_l = (p_s+p_t)/2$;
    calculate
    $g(p_l) \leftarrow p_l\dfrac{(1-\lambda_i)\,D_i}{B\log_2\left(1+\dfrac{p_l h_i}{N_0}\right)} + k(f_i^l)^2\lambda_i F_i - E_i$;
    if $g(p_l)\le 0$ **then**
      $p_s = p_l$;
    end
    if $g(p_l) > 0$ **then**
      $p_t = p_l$;
    end
  until $|p_t - p_s| \le \varepsilon$;
  $\bar{p}_i = (p_s+p_t)/2$;
end
return $\bar{p}_i$.

After the constraint C3 and C8 being solved, the new bounds for p_i is:

$$\hat{p}_{i,\min} = \max\left\{0, \frac{N_0\left(2^{\frac{(1-\lambda_i)D_i}{B\tau_i}} - 1\right)}{h_i}\right\} \tag{35}$$

and

$$\hat{p}_{i,\max} = \min\left\{p_{i,\max}, \bar{p}_i\right\} \tag{36}$$

Then, the optimal solution for p_i is:

$$p_i^* = \begin{cases} \hat{p}_{i,\max}, & if\ \hat{p}_{i,\max} \geq \hat{p}_{i,\min} \\ no\ feasible\ solution, & if\ \hat{p}_{i,\max} < \hat{p}_{i,\min} \end{cases} \tag{37}$$

D. Overall Algorithm for POSMU

The algorithm for solving **P3** is presented in **Algorithm 2**.

Algorithm 2: Algorithm for Solving P3

Input: $p_{i,\max}, p_{i,\min}, k, f_{i,\max}^l, \lambda_i, F_i, E_i, D_i, B, h_i, N_0$
Output: f_i^{c*}

initialize λ_i, f_i^l, p_i;
repeat
  calculate the optimal local computing frequency f_i^{l*} by calling (13);
  calculate the optimal local computing ratio λ_i^* by calling (26);
  calculate the optimal transmission power p_i^* by calling (37);
  update the objective function value:
  $$f\left(\lambda_i^*, p_i^*\right) \leftarrow \frac{\left(1-\lambda_i^*\right)F_i}{\tau_i - \frac{\left(1-\lambda_i^*\right)D_i}{B\log_2\left(1+\frac{p_i^* h_i}{N_0}\right)}};$$
until $f\left(\lambda_i^*, p_i^*\right)$ *is not changed*;
$f_i^{c*} \leftarrow f\left(\lambda_i^*, p_i^*\right)$;
return f_i^{c*}.

When an UE i has completely solved its optimization problem **P3**, the minimal MEC server frequency f_i^{c*} can be obtained. But the primal problem **P1** is still not solved. Because **P3** is less than **P1** a constraint C1, then the following **Algorithm 3** is proposed to get the maximal number of offloaded tasks N^* and satisfy C1.

POSMU contains two parts. The first part is calling the **Algorithm 2** to get the minimal f_i^{c*}s, and the second part is calculating the maximal number of offloaded UEs tasks N^* by calling **Algorithm 3**. Then the primal problem **P1** is solved completely.

V. Discussion for UAV Enabled MEC System

Due to their high mobility and flexible deployment, UAV can be used for the 5G communication. UAV can be used as a aerial platform to provide service for UEs, such as aerial base station, edge computing server for UEs, mobile-hubs for data collection, etc [42]. In order to realize the above usages, UAV-ground communication technology has made great progress. Different from the terrestrial communication, UAV-ground communication is applied in 3D space, and the uplink rate is affected by the distance between UAV and UE [43]:

Algorithm 3: Algorithm for Calculating the Maximal Number of Offloaded Tasks

Input: $f_i^{c*}, \forall i \in \mathcal{N}$, $f_{c,\max}$
Output: N^*

put all the f_i^{c*} in to set $\mathcal{F}$;
initialize $f_{total} \leftarrow 0, N_{user} \leftarrow 0$;
repeat
  get the minimal f_i^{c*} from $\mathcal{F}$, denoted by $f_i^{c,\min}$;
  $f_{total} \leftarrow f_{total} + f_i^{c,\min}$;
  $N_{user} \leftarrow N_{user} + 1$;
  $\mathcal{F} \leftarrow \mathcal{F} \backslash \left\{f_i^{c,\min}\right\}$
until $f_{total} \geq f_{c,\max}$;
$N^* \leftarrow N_{user}$;
return N^*.

$$r_i = B\log_2\left(1 + \frac{p_i h_0}{\left(H^2 + \left\|q - w_i\right\|^2\right)N_0}\right) \tag{38}$$

where h_0 represents the received power at the reference distance $d = 1$m. q is the UAV plane coordinates (x_{uav}, y_{uav}), and w_i is plane coordinates (x_i, y_i) of UE i. H is altitude of UAV.

According to the rate model, we find that if an UAV is hovering in the sky, it can be regarded as a MEC server with more distances from UEs compared with traditional MEC server nearby UEs. In our optimization model, the mobility of UAV is not considered. Therefore, the optimization problem **P1** can be used in UAV enabled MEC system with the changing of rate formula. Besides, each UE's plane coordinates should be preset and used by the rate formula. The solution idea of UAV enabled MEC system's optimization problem is the same as POSMU. Firstly, the primal problem is decoupled into multiple sub-problems, then relax and solve them based on BCD method. In order to get the maximal offloaded task number, **Algorithm 3** can be adopted. Therefore, our partial offloading strategy can be seamlessly adapted to UAV enabled MEC system.

VI. Numerical Results

A. Parameters Settings

The bandwidth B is 1×10^7Hz. α is set to 2.7 [35]. The channel gain for each UE h_i is set to -30dB. The noise power N_0 is -60dBm. The energy coefficient k is set to 10^{-26}.

We set the maximal transmission power $p_{i,max}$ for each UE to 0.4W. The maximal local computing frequency $f_{i,max}^l$

for each UE is 8×10^7Hz. The maximal computing frequency $f_{c,max}$ of the MEC server is set to 8×10^{11}Hz. Each UE has the initial energy E_i as 2J.

B. Comparison Schemes

According to POSMU, if f_i^c is smaller, the number of offloaded tasks N will be larger. Therefore, we can verify the optimality of POSMU by comparing the minimal value of f_i^c. Two schemes are proposed to evaluate the optimality. The first one is exhaustive algorithm (EA), which uniformly discretizes the domain of each variable in **P3** into 200 points. A large number of value combinations can be got by taking different discrete values of all the variables. By calculating the objective function values of these combinations one by one, the global optimal objective value of **P3** can be obtained. The second one is a monte carlo simulation (MCS) algorithm. In MCS, the domain of each variable in **P3** is discretized into 1000 points, and 10^7 combinations of these variables are sampled.

When illustrating advantages of POSMU, other two schemes are put forward. The first one is fixing the local computing frequency (FLCF) f_i^l as $0.5f_{i,\max}^l$ for each UE. The second one is fixing the transmission power (FTP) p_i as $0.2p_{i,\max}$ for each UE. Other parts of the two schemes are the same as that of POSMU. The delay for each UE is set to 2 seconds. The reason about using the two schemes for comparison is that few research work directly aims to maximize the offloaded task number. If POSMU is compared with other optimization schemes without the same objectives, it may cause some unfairness.

Simulation results compare two indexes, which are maximum number of offloaded task N^* and minimized MEC computing frequency f_i^{c*}. The reason why f_i^{c*} is simulated is that f_i^{c*} is the optimal value of subproblem **P3**, and it affects the maximum number of offloaded tasks N^*. Through the comparison of f_i^{c*}, it is easy to find out why the number of offloaded tasks N^* of POSMU is always the largest.

C. Optimality Evaluation

Each UE's task data D_i varies from 1×10^8bits to 4×10^8bits with the step as 0.3×10^8bits. The simulation results are presented in Fig.3. In the Fig.3, the optimized f_i^c for UE i is always the minimal compared with that of EA and MCS, which illustrates that POSMU can obtain an approximately global solution for the problem **P3**.

Besides, we also find that at some points such as 370, 340, etc the EA is not the best solution. This is because each of the variable domain is only discretized into 200 points, which may cause the optimal solution is not contained by this discretized points. The same reason is also suitable for MCS, which is because MCS samples 10^7 combinations of these variables but the total combinations number is 10^9.

D. Compared with FLCF and FTP

1) Changing the Data Size D_i: We randomly generate 2000 UEs tasks data with the formula $10^7 \times (x + 10 \times rand(1, 2000))$, where x is data size increment factor and varies from 30 to 60 with the step as 2 and $rand(1, 2000)$ means randomly generate 2000 random numbers belonging to $[0, 1]$. The simulation results are presented in Fig.4 and Fig.5. In addition, we set the bandwidth B to 2×10^7Hz for better illustration.

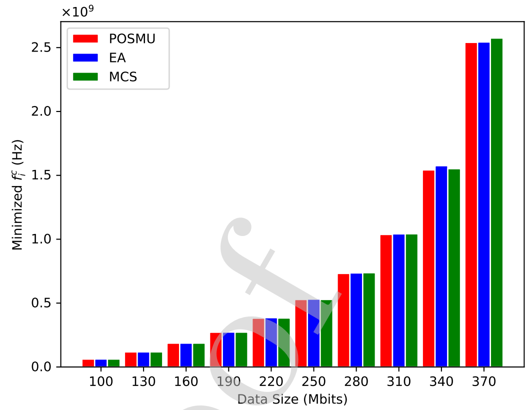

Fig. 3. Optimality of POSMU

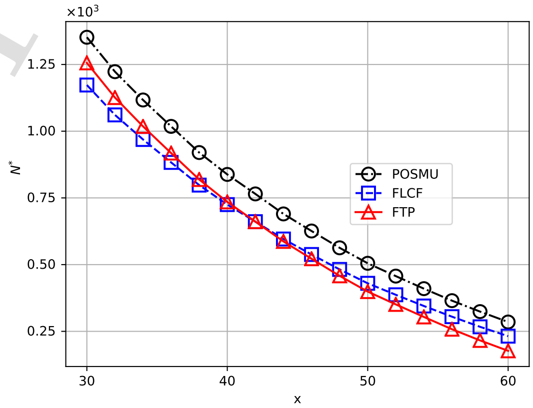

Fig. 4. Maximized Offloaded Tasks Number N^* Versus Data Size Increment Factor x

In the Fig.4, if the UEs tasks data size increases with x, the maximized offloaded tasks number N^* decreases. This is because if the data size increases, the computing frequency required from MEC server also increases, and then N^* will decrease. We also find that the N^* of POSMU is the maximal compared with FLCF and FTP.

In the Fig.5, if the UEs tasks data increases the total required computing frequency of all UEs' tasks increase rapidly especially when x is bigger than 55. Besides, the total required computing frequency optimized by POSMU is the least compared with FLCF and FTP, which makes the N^* of POSMU the maximal shown in Fig.4.

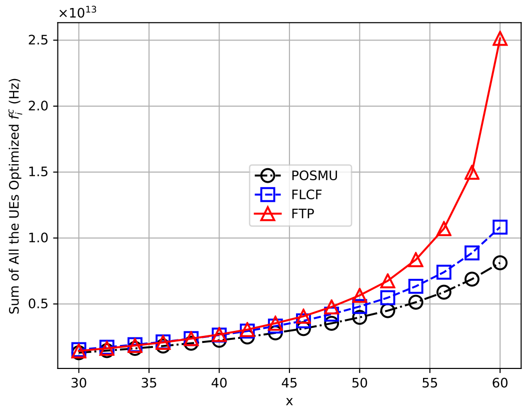

Fig. 5. Sum of All UEs' Optimized f_i^c Versus Data Size Increment Factor x

2) Changing the Maximal MEC Server Computing Frequency $f_{c,\max}$: $f_{c,\max}$ varies from 1×10^{11}Hz to 16×10^{11}Hz with the step as 1×10^{11}Hz. The bandwidth B is 2×10^7Hz. All the user data sizes are generated by the formula $10^7 \times (40 + 10 \times rand(1, 2000))$. Simulation results are presented in Fig.6.

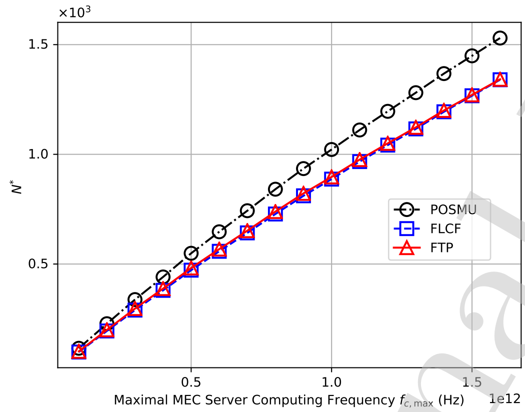

Fig. 6. Maximized Offloaded Tasks Number N^* Versus Maximal MEC Server Computing Frequency $f_{c,\max}$

According to the Fig.6, when the $f_{c,\max}$ increases, the maximal number of offloaded UEs tasks N^* also increases. This is because the increasing of $f_{c,\max}$ improves the computing frequency of MEC server, which further can offload more UEs tasks. In addition, the performance of POSMU is the best compared with FLCF and FTP. The differences between POSMU and other two schemes gradually grow with the increase of $f_{c,\max}$. This is because the performance of POSMU is always the best for each UE, i.e. each UE in POSMU has the minimal f_i^c compared with FLCF and FTP. Then, the same growth amount of $f_{c,\max}$ results more offloaded UEs tasks of POSMU compared with FLCF and FTP.

3) Changing the Bandwidth B: Each UE's task data size is generated by formula $10^7 \times (40 + 10 \times rand(1, 2000))$. Bandwidth B varies from 2×10^7Hz to 5×10^7Hz with the step as 0.2×10^7Hz. $f_{c,\max}$ is set to 8×10^{11}Hz. The simulation results are presented in Fig.7 and Fig.8.

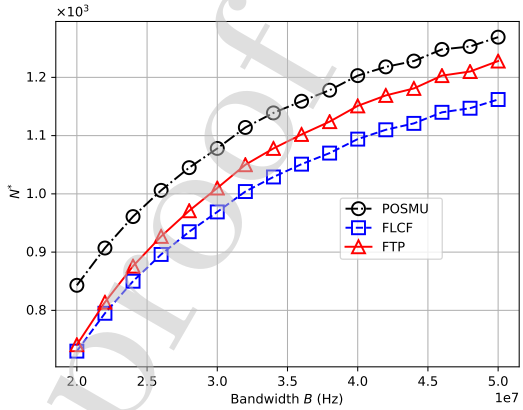

Fig. 7. Maximized Offloaded Tasks Number N^* Versus Bandwidth B

In the Fig.7, N^* increases with bandwidth B. This is because according to the transmission rate formulation (4), when the bandwidth B increases, the transmission rate grows, which reduces the data transmitting time and increases the time for MEC server processing. Thus, many UEs tasks can reduce their MEC server's computing frequency requirement and finally N^* becomes bigger. Besides, POSMU performs the best at all the simulation points.

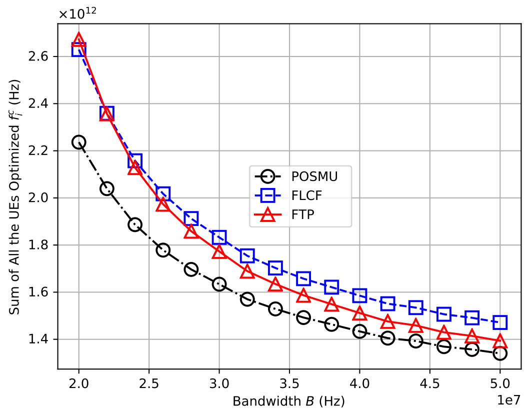

Fig. 8. Sum of All UEs' Optimized f_i^c Versus Bandwidth B

In the Fig.8, all the summed f_i^{c*} decreases as the bandwidth B increases. This is because if the bandwidth B increases, the time consumed for data transmission is reduced, which makes the processing time of MEC server increased. Then, the computing frequency f_i^c can be reduced. All the summed

f_i^{c*} of POSMU is the minimal compared with that of other schemes.

4) Changing the Latency Limitation: We change the value of τ_i from 1.0 second to 2.0 seconds with the step as 0.05 seconds, and assume that each task has the same latency limitation. The bandwidth B is set as 2×10^7Hz, and the UE's task data size is generated by $10^7\times(20+10\times rand(1,2000))$. The simulation results are shown in Fig.9 and Fig.10.

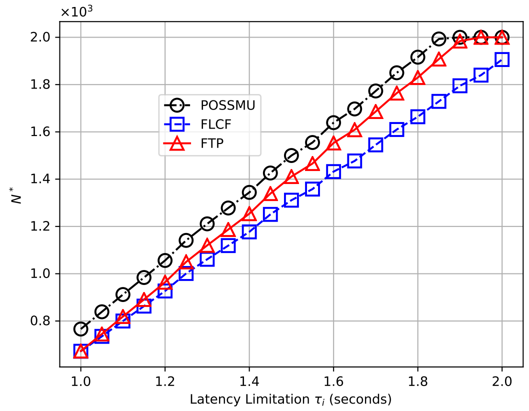

Fig. 9. Maximized Offloaded Tasks Number N^* Versus Latency Limitation τ_i

In the Fig.9, if the latency limitation τ_i increases, N^* grows accordingly. This is because when the τ_i increases, the processing time of MEC server also grows. Then the computing frequency f_i^c required to process the same data is reduced, which makes N^* increased. When the τ_i is bigger than 1.9, N^* is 2000 and not changed. This is because all the UEs tasks are offloaded to the MEC server. Besides, the performance of POSMU is the best among these algorithms.

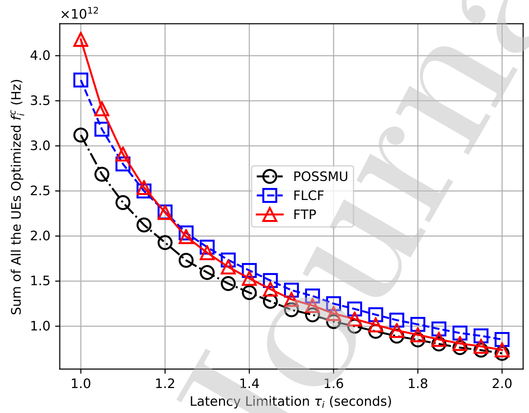

Fig. 10. Sum of All UEs' Optimized f_i^c Versus Latency Limitation τ_i

In the Fig.10, all the summed f_i^{c*} decreases with the growth of latency limitation τ_i. The reason is same as that presented in Fig.9. When the τ_i grows, the differences between POSMU and other two schemes become smaller. This is because when τ_i increases, each UE needs smaller MEC computing frequency f_i^c. In addition, because FLCF and FTP require more f_i^c than that of POSMU to process the same data, if the τ_i increases the reduction of f_i^c of FLCF and FTP is bigger than that of POSMU, thus the differences between POSMU and two schemes gradually become smaller. However, in any case all the summed f_i^{c*} of POSMU is the best among compared three schemes.

5) Changing the Maximal Local Computing Frequency $f_{i,\max}^l$: $f_{i,\max}^l$ varies from 6×10^7Hz to 14×10^7Hz with the step as 0.5×10^7Hz. The latency limitation τ_i is set to 1.5 seconds for all tasks. Bandwidth B is set to 2×10^7Hz. Each UE's task data size is generated by $10^7\times(20+10\times rand(1,2000))$. Simulation results are presented in Fig.11 and Fig.12.

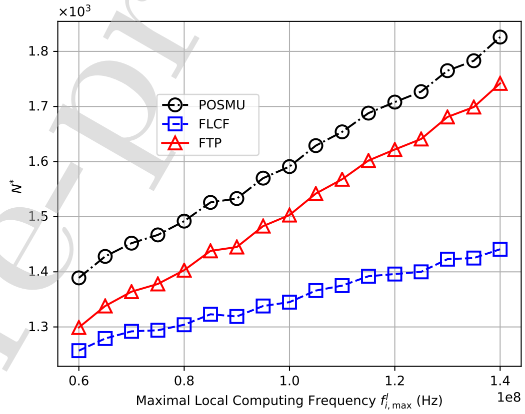

Fig. 11. Maximized Offloaded Tasks Number N^* Versus Maximal Local Computing Frequency $f_{i,\max}^l$

In the Fig.11, N^* grows with the increasing of $f_{i,\max}^l$, which is because if $f_{i,\max}^l$ increases the time consumption for local computing can be reduced. Then, the time for MEC server processing grows and the optimized MEC server frequency f_i^c is reduced. Thus, if $f_{i,\max}^l$ increases, N^* grows due to the reduction of f_i^c. Besides, the growth rate of FLCF is less than that of POSMU and FTP. This is because in FLCF the local computing frequency is fixed as $0.5f_{i,\max}^l$, while in POSMU and FTP the local computing frequency is optimized as an upper of f_i^l. Thus, the local computing frequency of FLCF is much less than that of POSMU and FTP, which results the slow growth of FLCF. In addition, N^* of POSMU is the largest at all the simulation points.

In the Fig.12, all the summed optimized f_i^c gradually decreases with the growth of $f_{i,\max}^l$. The reason is same as introduced in the Fig.11. Besides, descent rate of FLCF is the least because local computing frequency of FLCF is the least among three schemes. In addition, the summed optimized f_i^c of POSMU is the least.

6) Changing the Maximal Transmission Power $p_{i,\max}$: $p_{i,\max}$ varies from 0.1KW to 2.0KW with the step as 0.1KW. The latency limitation τ_i is set to 1.8 seconds. The maximal

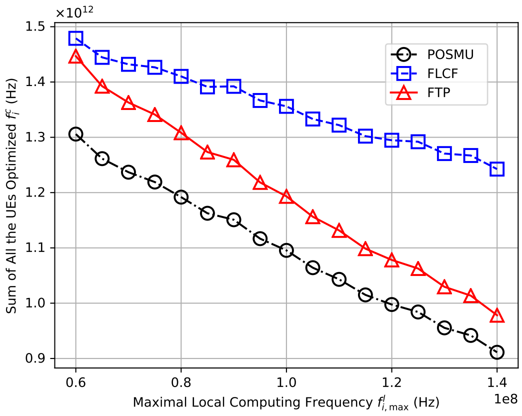

Fig. 12. Sum of All UEs' Optimized f_i^c Versus Maximal Local Computing Frequency $f_{i,\max}^l$

local computing frequency $f_{i,\max}^l$ is set to 8×10^7Hz. The UE's task data size is generated by $10^7 \times (20 + 10 \times rand(1, 2000))$. Simulation results are presented in Fig.13 and Fig.14.

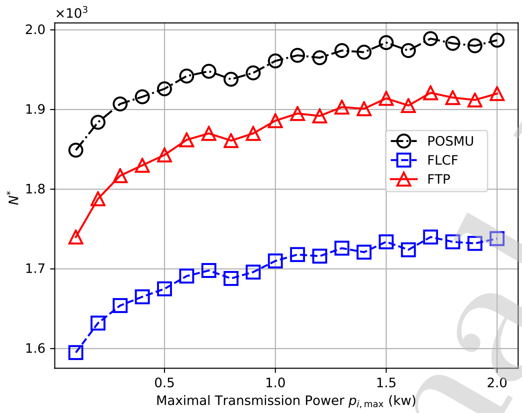

Fig. 13. Maximized Offloaded Tasks Number N^* Versus Maximal Transmission Power $p_{i,\max}$

In the Fig.13, N^* approximately grows with the increasing of $p_{i,\max}$. This is because when $p_{i,\max}$ increases the optimized transmission power p_i^* can also increases, which reduces the data transmission time. As a result, the time for MEC server processing becomes more, and then the MEC server computing frequency f_i^c can be reduced. Therefore, the total required MEC server's computing frequency is reduced and N^* grows.

In the Fig.14, all summed optimized f_i^c decreases with the growth of $p_{i,\max}$, and the reason is same as that introduced in the Fig.13. In addition, both in the Fig.13 and 14, the performance of POSMU is the best among three schemes.

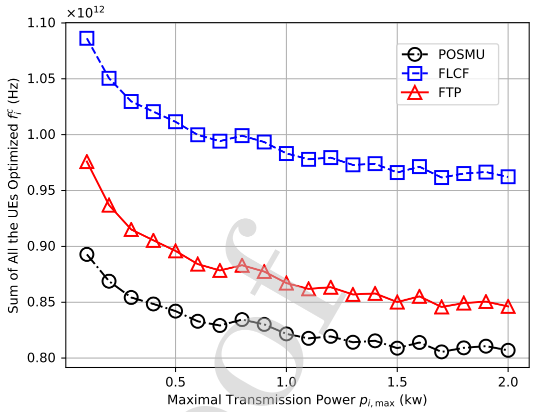

Fig. 14. Sum of All UEs' Optimized f_i^c Versus Maximal Transmission Power $p_{i,\max}$

VII. Conclusion

In this paper, we investigate a problem about maximizing the offloaded UEs tasks number in MEC system. Firstly, we formulate the problem as a MINLP, and then by analyzing the model, it is converted into N sub-problems, which are NLPs. In order to solve these N NLP sub-problems better, we relax the NLP problems and finally get the optimization sub-problem **P3**, which is analyzed and solved by using the BCD method, and finally an optimal scheme POSMU is proposed. In the evaluation section, the optimality of POSMU is illustrated, and also the performance of POSMU is compared with that of two schemes FLCF and FTP. All the evaluation results verify that POSMU is the best among compared schemes.

In the future, we will extend this work in the following aspects. Firstly, we will consider how to maximize the number of tasks to be offloaded in the scenario of multiple MEC servers with different characteristics. Secondly, due to the large number of UEs, it will cause serious interference. We will consider the communication interference between multiple UEs, and then optimize the maximum number of tasks that can be offloaded in this circumstances.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61772087), 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).

Declaration of interests

√The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

~~☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:~~

Author Statement

Qiang Tang: Writing-Reviewing and Editing, Software, Methodology

Lu Chang: Methodology, Data curation, Writing-Original draft preparation

Kun Yang: Conceptualization, Methodology

Kezhi Wang: Conceptualization, Methodology

Jin Wang: Investigation, Writing-Reviewing and Editing

Pradip Kumar Sharma: Conceptualization

[1] J. Wang, Y. Gao, W. Liu, W. Wu and S.-J. Lim, “An Asynchronous Clustering and Mobile Data Gathering Schema Based on Timer Mechanism in Wireless Sensor Networks,” in CMC: Computers, Materials and Continua, vol. 58, no. 3, pp. 711-725, 2019.
[2] J. Wang, C. Ju, Y. Gao, A. Kumar Sangaiah and G.-j. Kim, “A PSO based Energy Efficient Coverage Control Algorithm for Wireless Sensor Networks,” in CMC: Computers, Materials and Continua, vol. 56, no. 3, pp. 433-446, 2018.
[3] W. Li, Z. Chen, X. Gao, W. Liu and J. Wang, "Multimodel Framework for Indoor Localization Under Mobile Edge Computing Environment," in IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4844-4853, 2018.
[4] Q. Tang, M. Xie, K. Yang, Y. Luo, D. Zhou, Y. Song, “A Decision Function based Smart Charging and Discharging Strategy for Electric Vehicle in Smart Grid,” in Mobile Networks and Applications, vol. 24, no. 5, pp. 1722-1731, 2018.
[5] S. He, W. Zeng, K. Xie, H. Yang, M. Lai, X. Su, “PPNC: Privacy Preserving Scheme for Random Linear Network Coding in Smart Grid,” in KSII Transactions on Internet and Information Systems, vol. 11, no. 3, pp. 1510-1532, 2017.
[6] D. Cao, B. Zheng, B. Ji, Z. Lei, Chunhai Feng. “A Robust Distance-Based Relay Selection for Message Dissemination in Vehicular Network,” in Wireless Networks, DOI: 10.1007/s11276-018-1863-4, 2018.
[7] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," in IEEE Communications Surveys and Tutorials, vol. 19, no. 3, pp. 1628-1656, 2017.
[8] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang and W. Wang, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," in IEEE Access, vol. 5, no. 1, pp. 6757-6779, 2017.
[9] N. Abbas, Y. Zhang, A. Taherkordi and T. Skeie, "Mobile Edge Computing: A Survey," in IEEE Internet of Things Journal, vol. 5, no. 1, pp. 450-465, 2017.
[10] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," in IEEE Communications Surveys and Tutorials, vol. 19, no. 4, pp. 2322-2358, 2017.
[11] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," in IEEE Communications Surveys and Tutorials, vol. 19, no. 3, pp. 1657-1681, 2017.
[12] C. You, K. Huang and H. Chae, "Energy Efficient Mobile Cloud Computing Powered by Wireless Energy Transfer," in IEEE Journal on Selected Areas in Communications, vol. 34, no. 5, pp. 1757-1771, 2016.
[13] C. You, Y. Zeng, R. Zhang and K. Huang, "Asynchronous Mobile-Edge Computation Offloading: Energy-Efficient Resource Management," in IEEE Transactions on Wireless Communications, vol. 17, no. 11, pp. 7590-7605, 2018.
[14] S. E. Mahmoodi, R. N. Uma and K. P. Subbalakshmi, "Optimal Joint Scheduling and Cloud Offloading for Mobile Applications," in IEEE Transactions on Cloud Computing, vol. 7, no. 2, pp. 301-313, 2016.
[15] X. Hu, K. Wong and K. Yang, "Wireless Powered Cooperation-Assisted Mobile Edge Computing," in IEEE Transactions on Wireless Communications, vol. 17, no. 4, pp. 2375-2388, 2018.
[16] F. Guo, H. Zhang, H. Ji, X. Li and V. C. M. Leung, "An Efficient Computation Offloading

Management Scheme in the Densely Deployed Small Cell Networks With Mobile Edge Computing," in IEEE/ACM Transactions on Networking, vol. 26, no. 6, pp. 2651-2664, 2018.

[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," in IEEE Transactions on Wireless Communications, vol. 16, no. 9, pp. 5994-6009, 2017.

[18] Y. Dai, D. Xu, S. Maharjan and Y. Zhang, "Joint Computation Offloading and User Association in Multi-Task Mobile Edge Computing," in IEEE Transactions on Vehicular Technology, vol. 67, no. 12, pp. 12313-12325, 2018.

[19] L. Yang, H. Zhang, M. Li, J. Guo and H. Ji, "Mobile Edge Computing Empowered Energy Efficient Task Offloading in 5G," in IEEE Transactions on Vehicular Technology, vol. 67, no. 7, pp. 6398-6409, 2018.

[20] J. Wang, W. Wu, Z. Liao, A. K. Sangaiah and R. Simon Sherratt, "An Energy-Efficient Off-Loading Scheme for Low Latency in Collaborative Edge Computing," in IEEE Access, vol. 7, no. 1, pp. 149182-149190, 2019.

[21] H. Mazouzi, K. Boussetta, N. Achir, “Maximizing mobiles energy saving through tasks optimal offloading placement in two-tier cloud: A theoretical and an experimental study,” in Computer Communications, vol 144, no. 1, pp. 132-148, 2018.

[22] J. Ren, G. Yu, Y. Cai, Y. He and F. Qu, "Partial Offloading for Latency Minimization in Mobile-Edge Computing," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, pp. 1-6, 2017.

[23] J. Ren, G. Yu, Y. Cai and Y. He, "Latency Optimization for Resource Allocation in Mobile-Edge Computation Offloading," in IEEE Transactions on Wireless Communications, vol. 17, no. 8, pp. 5506-5519, 2018.

[24] S. Yu, R. Langar, X. Fu, L. Wang and Z. Han, "Computation Offloading With Data Caching Enhancement for Mobile Edge Computing," in IEEE Transactions on Vehicular Technology, vol. 67, no. 11, pp. 11098-11112, 2018.

[25] A. Samanta and Z. Chang, "Adaptive Service Offloading for Revenue Maximization in Mobile Edge Computing With Delay-Constraint," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3864-3872, 2019.

[26] Y. Wang, M. Sheng, X. Wang, L. Wang and J. Li, "Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling," in IEEE Transactions on Communications, vol. 64, no. 10, pp. 4268-4282, 2016.

[27] Q. Tang, H. Lyu, G. Han, J. Wang, K. Wang, Partial offloading strategy for mobile edge computing considering mixed overhead of time and energy, Neural Computing and Applications, DOI: 10.1007/s00521-019-04401-8, 2019.

[28] R. Wang, Y. Cao, A. Noor, T. A Alamoudi, and R. Nour, “Agent-enabled task offloading in UAV-aided mobile edge computing,” in Computer Communications, vol. 149, no. 1, pp. 324-331, DOI: 10.1016/j.comcom.2019.10.021, 2019.

[29] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao and D. Chen, "Joint Resources and Workflow Scheduling in UAV-Enabled Wirelessly-Powered MEC for IoT Systems," in IEEE Transactions on Vehicular Technology, vol. 68, no. 10, pp. 10187-10200, 2019.

[30] H. Mei, K. Wang, D. Zhou and K. Yang, "Joint Trajectory-Task-Cache Optimization in UAV-Enabled Mobile Edge Networks for Cyber-Physical System," in IEEE Access, vol. 7, pp. 156476-156488, 2019.

[31] F. Zhou, Y. Wu, R. Q. Hu and Y. Qian, "Computation Rate Maximization in UAV-Enabled Wireless-Powered Mobile-Edge Computing Systems," in IEEE Journal on Selected Areas in Communications, vol. 36, no. 9, pp. 1927-1941, 2018.

[32] Z. Yang, C. Pan, K. Wang and M. Shikh-Bahaei, "Energy Efficient Resource Allocation in UAV-Enabled Mobile Edge Computing Networks," in IEEE Transactions on Wireless Communications, vol. 18, no. 9, pp. 4576-4589, 2019.

[33] X. Hu, K. Wong, K. Yang and Z. Zheng, "UAV-Assisted Relaying and Edge Computing: Scheduling and Trajectory Optimization," in IEEE Transactions on Wireless Communications, vol. 18, no. 10, pp. 4738-4752, 2018.

[34] Y. Wang, M. Sheng, X. Wang, L. Wang and J. Li, "Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling," in IEEE Transactions on Communications, vol. 64, no. 10, pp. 4268-4282, 2016.

[35] O. Muñoz, A. Pascual-Iserte and J. Vidal, "Optimization of Radio and Computational Resources for Energy Efficiency in Latency-Constrained Application Offloading," in IEEE Transactions on Vehicular Technology, vol. 64, no. 10, pp. 4738-4755, 2014.

[36] X. Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," in IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 4, pp. 974-983, 2014.

[37] Y. Zhang, J. He and S. Guo, "Energy-Efficient Dynamic Task Offloading for Energy Harvesting Mobile Cloud Computing," 2018 IEEE International Conference on Networking, Architecture and Storage (NAS), Chongqing, pp. 1-4, 2018.

[38] Y. L. Lee, T. C. Chuah, J. Loo and A. Vinel, "Recent Advances in Radio Resource Management for Heterogeneous LTE/LTE-A Networks," in IEEE Communications Surveys \& Tutorials, vol. 16, no. 4, pp. 2142-2180, 2014.

[39] Y. Hao, M. Chen, L. Hu, M. S. Hossain and A. Ghoneim, "Energy Efficient Task Caching and Offloading for Mobile Edge Computing," in IEEE Access, vol. 6, pp. 11365-11373, 2018.

[40] Xu, Yangyang , and W. Yin, "A Block Coordinate Descent Method for Regularized Multiconvex Optimization with Applications to Nonnegative Tensor Factorization and Completion," in SIAM Journal on Imaging Sciences, vol. 6, no. 3, pp. 1758-1789, 2013.

[41] S.Boyd and L. Vandenberghe, Convex Optimization. Cambridge, MA, USA: Cambridge Univ. Press, 2004.

[42] Q. Wu, L. Liu and R. Zhang, "Fundamental Trade-offs in Communication and Trajectory Design for UAV-Enabled Wireless Network," in IEEE Wireless Communications, vol. 26, no. 1, pp. 36-44, 2019.

[43] Y. Du, K. Wang, K. Yang and G. Zhang, "Energy-Efficient Resource Allocation in UAV Based MEC System for IoT Devices," 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, pp. 1-6, 2018.