# Latency-Optimal Task Offloading for Mobile-Edge Computing System in 5G Heterogeneous Networks

Guoxuan Chi, Yumei Wang, Xiang Liu, Yiming Qiu
School of Information and Communication Engineering
Beijing University of Posts and Telecommunications
Beijing, China, 100876
Email: {chiguoxuan12345, ymwang}@bupt.edu.cn

***Abstract*—Mobile edge computing (MEC) is an emerging technology to improve the quality of computation experience for mobile devices. As a promising paradigm to deal with latency-sensitive and computation-intensive tasks, it provides cloud computing capabilities in close proximity to mobile devices in the fifth-generation (5G) networks. As the radio and computational resources are both limited in 5G networks, reducing system latency by task scheduling and resource allocation has gained renewed interests. To minimize the weighted-sum latency of all users in multi-user MEC system, we formulate an optimization problem based on partial offloading strategy. Since the optimization problem is NP-hard, we transform it into a piecewise convex problem and get the latency-optimal offloading strategy using the sub-gradient method. We further put forward a simplified algorithm which can achieve close-to-optimal performance in linear time. Our proposed strategies are verified by numerical results, which indicate that our algorithms significantly reduce the weighted-sum latency compared with other baseline strategies.**

**Keywords—Mobile-edge computing (MEC); task offloading; 5G networks; latency-optimal.**

## I. INTRODUCTION

The explosive popularity of smart mobile devices accelerates the advent of many new mobile applications and services (e.g., virtual reality, augmented reality and ultra-high-definition video streaming), most of which are latency-sensitive and computation -intensive [1][2]. Although Cloud Computing and Mobile Cloud Computing (MCC) both tried to reduce the computational workload of mobile devices by migrating tasks to cloud servers, central network congestion often leads to serious delay jitter [3]. In addition, the privacy and security have also become the key problems of MCC system [4]. Compared with two paradigms mentioned above, MEC can reduce system latency as well as energy consumption of mobile devices, thereby improving the quality of experience (QoE) for users [5]. By deploying servers on the network edges (e.g. base stations and access points), MEC provides computing capabilities for end-users and Internet Content Providers (ICP) [2]. The short distance between servers and mobile devices can hardly result in additional delay, which makes MEC one of the most promising paradigms to improve the performance of 5G heterogeneous networks.

Several studies have addressed the computation offloading and resource allocation problems on both single-user [6]-[8] and multi-user MEC systems [9]-[13] in recent years. In [6], the authors have formulated a task assignment problem by constructing dependency relationships of transactions, which can be solved by task graph scheduling. By means of queuing theory and binary offloading (i.e., a task can be executed either on a mobile device or on the edge cloud servers), the task scheduling strategy in [7] significantly reduces the system latency. In [10], the computation offloading strategy has been further developed to minimize the weighted-sum energy consumption of mobile devices. In addition, a stochastic task arrival model based on the Lyapunov optimization [11] has been proposed to solve the energy-latency tradeoff problem for a multi-user MEC system. Based on binary offloading, an energy-efficient offloading strategy for MEC system in 5G networks has been proposed in [12]. Most aforementioned works focused on the binary offloading model and the optimization of energy consumption. However, a latency-optimal strategy based on partial offloading model is also an urgent need of the near-future 5G wireless networks.

In this paper, we consider a multi-user MEC system in 5G networks with multiple independent tasks. Under the radio and computational resource constraints, the system needs to allocate all the resources efficiently. We optimize the offloading proportion and resource allocation for MEC systems through partial offloading (i.e., every task can be arbitrarily divided for local and remote executions), which improves the parallelism of MEC systems. However, latency-optimal partial offloading can hardly be achieved by basic optimization methods since the resource allocation and task segmentation are tightly coupled. In addition, network structure makes the optimization problem rather difficult to be solved. Therefore, it's a challenging work to find out an effective way to achieve the lowest latency. By analyzing the objective function, we transform the original problem into a piecewise convex problem, which can be solved by the sub-gradient method. To simplify the strategy and reduce the scheduling time, a linear-time-complexity algorithm based on a common scenario is further derived.

The rest of this paper is organized as follows. In Section II, we introduce the system model which includes computation and communication models. The latency optimization problem based on resource allocation and data segmentation is formulated in Section III. Problem-solving methods are discussed in Section IV, where both optimal and close-to-optimal algorithms are proposed. Numerical results are shown in Section V and conclusions are drawn in Section VI.

This work is sponsored by the National College Innovation Program of Beijing University of Posts and Telecommunications, and Huawei Cooperative Research Project (No. YBN2016110032).

978-1-5386-6355-4/18/$31.00 ©2018 IEEE
Authorized licensed use limited to: Tsinghua University. Downloaded on September 02,2024 at 02:51:18 UTC from IEEE Xplore. Restrictions apply.


Fig. 1. MEC system in 5G networks

## II. System Model

A multi-user MEC system in 5G networks shown in Fig. 1 consists of a single macro base station (MBS) that covers a macrocell, multiple small base stations (SBS) that cover different microcells, and multiple mobile devices with computation-intensive applications. SBS and MBS are equipped with small edge servers (SES) and large edge servers (LES) respectively. SBSs are connected with MBS through optical cables. Mobile devices in microcells are connected with SBS, while devices outside these microcells are connected with MBS. The MEC system will re-arrange computation offloading strategy at intervals and transmit different data segments to LES, SES and local devices for parallel computing. Finally, all processed data will be aggregated on user's mobile devices.

### A. Task Model

Consider an MEC system including $N$ mobile devices. Denote the task set as $\mathcal{T} = \{T_1, T_2, \dots, T_N\}$, where $T_d$ represents a task requested by device $d$. Each task is characterized by a three-tuple of parameters $\langle S_d, C_d, \beta_d \rangle$, where $S_d$ (in bits) and $C_d$ (in CPU cycles/bit) denote size and workload of the task respectively, while $\beta_d \in (0, +\infty)$ indicates the data growth ratio (i.e., one-bit raw data will be processed into $\beta_d$ bits).

### B. Computation Model

Suppose there are $M$ SBSs denoted by a set $\mathcal{M} = \{1, 2, \dots, M\}$. Let $V_d$, $V_m$ and $V_c$ (in CPU cycles/sec) be the computing capacity of device $d$, SES $m$ and LES respectively. We define *edge* as a microcell or a single device connected with MBS, and $V_e$ as its computing capacity. In a multi-user MEC system, servers are used by many devices at the same time. Assume $V_m^d$ denotes the computing capacity of SES $m$ allocated to device $d$, while $V_c^e$ denotes the computing capacity of LES allocated to edge $e$. Since we can't get the cooperative computing capacity of a microcell directly, $V_e$ is estimated by means of context-awareness. We suppose that MEC servers are equipped with multi-core CPUs and can be virtualized into several virtual machines (VMs), hence the concurrent of multi-tasks is feasible.

### C. Communication Model

Assume that in an OFDMA system, all BSs operate in the same frequency band with a bandwidth $W$. The spectrum is divided into $K$ orthogonal sub-channels denoted as a set $\mathcal{K} = \{1, 2, \dots, K\}$. The bandwidth of each sub-channel is identical. Denote $\alpha_k^d = \{0, 1\}$ as a state indicator of sub-channel $k$, that is, if sub-channel $k$ is allocated to device $d$, then $\alpha_k^d = 1$, otherwise, $\alpha_k^d = 0$. So the data transmission rate can be expressed as

$$r_d = \sum_{k=1}^{K} \alpha_k^d \cdot \frac{W}{K} log_2 \left(1 + \frac{P_k \cdot \left|g_k^d\right|^2}{N_0}\right), \quad (1)$$

where $P_k$ and $N_0$ denote the transmission power of BS in channel $k$ and the variance of additive white Gaussian noise (AWGN), while $g_k^d$ indicates the channel gain of device $d$ on channel $k$. Since $g_k^d$ is a random variable, data transmission rate $r_d$ is also a random variable. According to [14], we get the expected value to evaluate transmission rate $r_d$. For ease of notion, we define $\rho_d = \frac{1}{K} \cdot \sum_{k=1}^{K} \alpha_k^d$ which represents the channel occupancy of device $d$.

Transmission rates between BSs are considered as $\varphi_0$. Communication resource constraints are not considered therein because of the large transmission capacity of optical cables.

### D. Two-step Optimization Model

For device $d$ in a microcell, both resource allocation and task segmentation strategies are sequentially executed twice. The first and the second segmentation proportions are denoted by $\lambda_e \in [0,1]$ and $\mu_d \in [0,1]$. Define **Step 1** and **Step 2** as two optimization strategies executed at MBS and SBS respectively. The detailed procedure can be described as follows:

- **Step 1:** After receiving $S_d$ bits raw data from ICP, MBS processes $\lambda_e S_d$ bits data on LES and transmits the remaining $(1-\lambda_e)S_d$ bits raw data to SBS. Then, $\beta_d \lambda_e S_d$ bits processed data will also be transmitted to SBS.
- **Step 2:** SBS receives $(1-\lambda_e)S_d$ bits raw data from MBS, and then processes $(1-\lambda_e)\mu_d S_d$ bits data on SES, while $(1-\lambda_e)(1-\mu_d)S_d$ bits are transmitted to device $d$ for local computing. Then, $\beta_d(1-\lambda_e)\mu_d L_d$ bits processed data will be transmitted to device $d$.

In particular, for device $i$ which connects to MBS directly, the optimization strategy will only be executed once. That means $\lambda_i S_i$ bits data will be processed on LES, while the other part will be transmitted to device and then processed locally.

TABLE I. Latency Expressions

| No. | Latency Expressions | Sub-processes |
|---|---|---|
| ① | $L_c^c = \sum_{d=1}^{n} \left(\frac{C_d S_d}{V_c^e}\right) \lambda_e$ | Process raw data on LES |
| ② | $L_c^t = \sum_{d=1}^{n} \left(\frac{\beta_d S_d}{R_e}\right) \lambda_e$ | Transmit processed data to edge |
| ③ | $L_e^t = \sum_{d=1}^{n} \left(\frac{S_d}{R_e}\right) (1-\lambda_e)$ | Transmit raw data to the edge |
| ④ | $L_e^c = \sum_{d=1}^{n} \left(\frac{C_d S_d}{V_e}\right) (1-\lambda_e)$ | Process raw data on the edge |
| ⑤ | $L_m^c = \frac{C_d S_d}{V_m^d} (1-\lambda_e)\mu_d$ | Process raw data on SES |
| ⑥ | $L_m^t = \frac{\beta_d S_d}{\rho_d R_d} (1-\lambda_e)\mu_d$ | Transmit processed data to device |
| ⑦ | $L_d^t = \frac{S_d}{\rho_d R_d} (1-\lambda_e)(1-\mu_d)$ | Transmit raw data to device |
| ⑧ | $L_d^c = \frac{C_d S_d}{V_d} (1-\lambda_e)(1-\mu_d)$ | Process raw data on device |

The latency expressions of eight sub-processes in Fig. 2 are shown in TABLE I.

Authorized licensed use limited to: Tsinghua University. Downloaded on September 02,2024 at 02:51:18 UTC from IEEE Xplore. Restrictions apply.

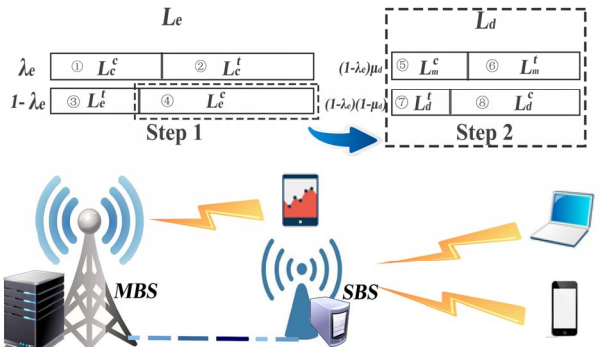



Fig. 2. Two-step Optimization Model

## III. Problem Formulation

In this section, we formulate the optimization problem by analyzing the concurrency relations of different sub-processes mentioned in Section II. Since mobile devices can hardly receive raw data and processed data at the same time, local computing part and cloud computing part can be transmitted at any moment while not simultaneously. Therefore, the latency expression of edge $e$ can be shown as

$$L_e = \begin{cases} max\{L_e^t + L_e^c\ ,\ L_c^t + L_c^c\}, & L_e^t \le L_c^c, \\ L_e^t + max\{L_c^t\ ,\ L_e^c\}, & L_e^t > L_c^c. \end{cases} \tag{2}$$

Since edge $e$ could be a microcell, we should further consider the optimization of **Step 2**. According to TABLE I. , the latency for cooperative computing in microcell $m$ can be shown as

$$L_d = \begin{cases} max\{L_d^t + L_d^c\ ,\ L_m^t + L_m^c\}, & L_d^t \le L_m^c, \\ L_d^t + max\{L_m^t\ ,\ L_d^c\}, & L_d^t > L_m^c. \end{cases} \tag{3}$$

**Step 1** and **Step 2** both try to minimize the weighted-sum latency under radio and computational resource constraints, which indicates that two optimization problems can be formulated in the same form. Denote a positive weight factor set $\{w_n\}$ which satisfies $\sum_{n=1}^{N} w_n = 1$. Therefore, the optimization problem is shown as

$$min\ \sum_{n=1}^{N} w_n L_n, \tag{4a}$$
$$\mathbf{s.t.}\ \sum_{n=1}^{N} \rho_n \le 1, \quad \rho_n > 0, \tag{4b}$$
$$\sum_{n=1}^{N} V_n \le V, \quad V_n \ge 0, \tag{4c}$$

where (4b) and (4c) are the radio and computational resource constraints respectively. For ease of notion, the discrete variable $\rho_n$ should be converted into a continuous variable by relaxation and rounding method.

We formulate the optimization problems of two steps by substituting (2) and (3) into (4a) respectively. The weighted-sum latency can be minimized by solving the two problems.

## IV. Problem Solving

As we can see from Section III, the problem-solving methods of two steps are exactly the same and the optimization of **Step 1** can be derived from **Step 2**. Thus, we mainly focus on **Step 2**, which is relatively easy to be illustrated. Both optimal and close-to-optimal strategies are discussed in this section.

### A. Problem Transformation

The piecewise expression of $L_d$ is quite complicated with three variables (i.e., $\mu_d, \rho_d$ and $V_m^d$) coupled with one another, so the optimization problem (4a) can hardly be solved directly. Consider a special case

$$\begin{cases} L_d^t = L_m^c, \\ L_d^c = L_m^t, \end{cases} \tag{5}$$

which means that the equivalent processing capability of device $d$ equals to its transmission capacity (i.e., $\mathrm{R}_d\rho_d C_d = \sqrt{\beta_d V_m^d V_d}$). Thus, two special segmentation proportions are solved as

$$\begin{cases} \mu_d^1 = \frac{V_m^d}{V_m^d + \mathrm{R}_d\rho_d C_d}, \\ \mu_d^2 = \frac{\mathrm{R}_d\rho_d C_d}{\beta_d V_d + \mathrm{R}_d\rho_d C_d}. \end{cases} \tag{6}$$

The optimization problem can be discussed in two cases.

- *Case A*: $\mu_d^1 \le \mu_d^2$ (i.e., $\mathrm{R}_d\rho_d C_d \ge \sqrt{\beta_d V_m^d V_d}$).

  When $\mu_d \in [0, \mu_d^1)$, $L_d^t > L_m^c$ and $L_d^c > L_m^t$. We get $L_d = L_d^t + L_d^c$, which decreases with $\mu_d$. When $\mu_d \in (\mu_d^2, 1]$, $L_m^t > L_d^c$ and $L_m^c > L_d^t$. We get $L_d = L_m^t + L_m^c$ which increases with $\mu_d$. When $\mu_d \in (\mu_d^1, \mu_d^2)$, we get $L_d = max\{L_d^t + L_d^c, L_m^t + L_m^c\}$. As $L_d^t + L_d^c$ decreases with $\mu_d$ while $L_m^t + L_m^c$ increases with $\mu_d$, the minimum value can be achieved when $L_d^t + L_d^c = L_m^t + L_m^c$.

- *Case B*: $\mu_d^1 > \mu_d^2$ (i.e., $R_d\rho_d C_d < \sqrt{\beta_d V_m^d V_d}$).

  When $\mu_d \in [0, \mu_d^2)$, $L_d^t > L_m^c$ and $L_d^c > L_m^t$. We get $L_d = L_d^t + L_d^c$ which decreases with $\mu_d$. When $\mu_d \in (\mu_d^2, 1]$, we get $L_d = max\{L_m^t + L_m^c,\ L_m^t + L_d^t\}$, which increase with $\mu_d$. So the minimum value can be achieved when $\mu_d = \mu_d^2$.

Therefore, the data segmentation proportion is shown as

$$\mu_d = \begin{cases} \frac{V_m^d(V_d + \mathrm{R}_d\rho_d C_d)}{(1+\beta_d)V_d V_m^d + \mathrm{R}_d\rho_d C_d(V_d + V_m^d)}, & \mu_d^1 \le \mu_d^2, \\ \frac{\mathrm{R}_d\rho_d C_d}{\beta_d V_d + \mathrm{R}_d\rho_d C_d}, & \mu_d^1 > \mu_d^2. \end{cases} \tag{7}$$

Substitute (7) into (3), the latency expression can be written as

$$\hat{L}_d = \begin{cases} \frac{\beta_d V_m^d(V_d + \mathrm{R}_d\rho_d C_d) + V_d(\mathrm{R}_d\rho_d C_d)^2 S_d(1-\lambda_e)}{(1+\beta_d)V_d V_m^d + (\mathrm{R}_d\rho_d C_d)^2(V_d + V_m^d)}, & \rho_d \ge \frac{\sqrt{\beta_d V_m^d V_d}}{\mathrm{R}_d C_d}, \\ \frac{\beta_d S_d(1-\lambda_e)}{\beta_d V_d + \mathrm{R}_d\rho_d C_d}, & \rho_d < \frac{\sqrt{\beta_d V_m^d V_d}}{\mathrm{R}_d C_d}. \end{cases} \tag{8}$$

### B. Optimal Strategy

The transformed latency expression (8) is continuous but non-differential at piecewise point so the classical KKT conditions can't be applied. We substitute (8) into (4a) and then solve the optimization problem of **Step 2** by means of the sub-gradient method. Sub-gradient function is shown as

$$\boldsymbol{g} = \begin{cases} \boldsymbol{\partial}\left(\sum_{d=1}^{n} w_d \hat{L}_d\right), \\ \boldsymbol{\partial}\left(\sum_{d=1}^{n} \rho_d\right), \\ \boldsymbol{\partial}\left(\sum_{d=1}^{n} V_m^d\right), \end{cases} \tag{9}$$

Authorized licensed use limited to: Tsinghua University. Downloaded on September 02,2024 at 02:51:18 UTC from IEEE Xplore. Restrictions apply.

where $\boldsymbol{\partial}\left(\sum_{d=1}^{n} w_d \hat{L}_d\right)$ is a piecewise expression. According to non-differential convex optimization theory, the problem can be solved by iteration expression shown as

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \phi_n \boldsymbol{g}^{(n)}, \tag{10}$$

where $\boldsymbol{x}$ denotes the resource allocation strategy. The whole procedure of sub-gradient algorithm is presented in Algorithm 1.

**Algorithm 1** Sub-gradient algorithm

1: **Input** $V_d = [V_1, \dots, V_N]$, $V_{SES} = [V_{S1}, \dots, V_{SM}]$, $P_{BS} = P_0$, $T_d = [T_1, \dots, T_N]$, $R_d = [R_1, \dots, R_N]$, $\epsilon = \epsilon_0$ and $n = 0$.
2: **Output** $\boldsymbol{x}^{(n)}$ and $\boldsymbol{\mu} = [\mu_1, \dots, \mu_n]$.
3: **Do**
4: Set $F_{(old)} = F_{(new)}$.
5: Update $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \phi_n \boldsymbol{g}^{(n)}$.
6: Update $F_{(new)} = \sum_{d=1}^{n} w_d \hat{L}_d$.
7: Update $\boldsymbol{\mu}$ according to (7).
8: Update $\boldsymbol{g}^{(n)}$ according to (9).
9: Set $n = n + 1$.
10: **While** $F_{(old)} - F_{(new)} \geq \epsilon_0$.

Although it is an algorithm with polynomial time complexity $\mathcal{O}\left(\frac{1}{\epsilon_0{}^2}\right)$, $\epsilon_0$ has to be an infinitesimal value to achieve the optimal latency. That would cause a tremendous execution delay, which is completely intolerable in practical scenarios.

### C. Close-to-optimal Strategy

In practical scenarios, locations of mobile devices and their requested contents are constantly changing so the resource allocation and data segmentation are dynamic. The algorithm needs to be executed every few seconds. Therefore, we try to simplify Algorithm 1 by considering a common case.

Since computation-intensive applications usually take a long execution time, some of the new features in 5G (e.g. higher bandwidth, massive MIMO and beamforming technology) guarantee that the data transmission rate is far greater than processing rate, which means $L_d^t \ll L_d^c$ and $L_e^t \ll L_e^c$. Therefore, $\mu_d^1 \leq \mu_d^2$ is always true. Besides, the equation $L_d^t + L_d^c = L_m^t + L_m^c$ can also be approximatively written as $L_d^c = L_m^t + L_m^c$ so that a new approximate segmentation proportion can be calculated. Substitute the new proportion into (3) and $\bar{L}_d$ can be shown as

$$\bar{L}_d = \frac{(\mathrm{R}_d \rho_d C_d + \beta_d V_m^d) T_d (1-\lambda_e)}{\beta_d V_d V_m^d + \mathrm{R}_d \rho_d C_d (V_d + V_m^d)}, \tag{11}$$

which is a differential function. Thus, the convex problem can be solved by classic KKT condition. Construct a Lagrange function of $\sum_{d=1}^{n} w_d \bar{L}_d$ shown as

$$F_d = \sum_{d=1}^{n} w_d \bar{L}_d + \delta\left(\sum_{d=1}^{n} \rho_d - 1\right) + \varepsilon\left(\sum_{d=1}^{n} V_m^d - V_m\right), \tag{12a}$$

$$\text{s.t.} \quad \sum_{d=1}^{n} \rho_d \leq 1, \rho_d > 0, \quad \delta\left(\sum_{d=1}^{n} \rho_d - 1\right) = 0, \tag{12b}$$

$$\sum_{d=1}^{n} V_m^d \leq V_m, V_m^d \geq 0, \ \varepsilon\left(\sum_{d=1}^{n} V_m^d - V_m\right) = 0, \tag{12c}$$

where $\delta \geq 0$ and $\varepsilon \geq 0$ are Lagrange multipliers associated with radio and computational resource constraints respectively.

By computing the partial derivatives of (12a) and let them equal to zero, the close-to-optimal resource allocation strategy can be expressed as

$$\begin{cases} \rho_d = \dfrac{\left(\sqrt{\frac{\beta_d \sum_{d=1}^{n} w_d R_d}{\delta}} - \beta_d V_d\right) V_m^d}{R_d C_d (V_d + V_m^d)}, \\ V_m^d = \dfrac{\left(\sqrt{\frac{\beta_d}{\varepsilon}} - V_d\right) R_d \rho_d C_d}{\beta_d V_d + R_d \rho_d C_d}. \end{cases} \tag{13}$$

Since $\rho_d$ and $V_m^d$ are still coupled, we substitute their expressions into each other. After transforming (12b) and (12c) into equality constraints, we get the optimal resource allocation of **Step 2** by solving a linear system of equations. Substitute (13) into (7), the offloading proportion $\mu_d$ can be solved.

The whole procedure of close-to-optimal strategy is presented in Algorithm 2.

**Algorithm 2** Close-to–optimal offloading algorithm

**Step 1**
Update $V_d, T_d, R_e$ and $R_d$ by using context-awareness
Calculate $V_c^e$, $\rho_e$ and $\lambda_e$.
**Step 2**
Calculate $V_m^d$, $\rho_d$ and $\mu_d$ according to (13).
Calculate $L_d$ according to (11).
Set $V_e = \sum_{d=1}^{n} S_d / max\{L_d\}$.

Since explicit expressions have been given in (7) and (13), the time complexity of Algorithm 2 is $\mathcal{O}(n)$, which is determined by the number of mobile devices.

## V. Numerical Results

To verify our proposed two strategies, we first compare weighted-sum latency between different strategies and then evaluate resource allocation between mobile devices with different computing capacities. To reduce the randomness, we perform 500 independent repeated trails. Major simulation parameters are listed in TABLE II.

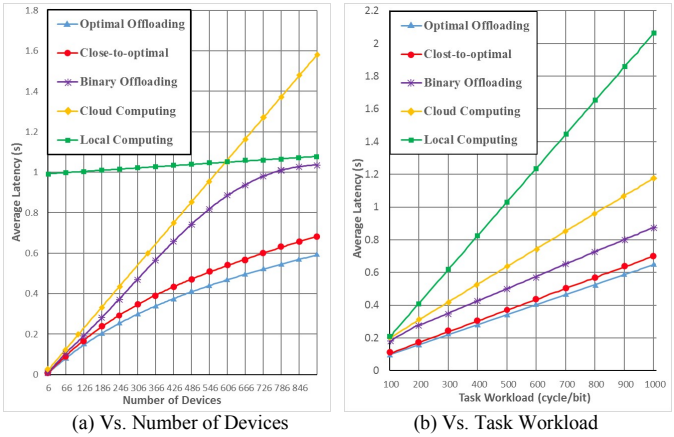
TABLE II. Mayjor Simulation Parameters

| Parameters | Value | Parameters | Value |
|---|---|---|---|
| Macrocell radius | 800 m | MBS power | 44 dBm |
| Microcell radius | 150 m | SBS power | 34 dBm |
| Optical-cable rate | 8000 Mbit/s | LES capacity | $10^{12}$ cycles/s |
| Noise power density | -174 dB/Hz | SES capacity | $10^{11}$ cycles/s |
| Pathloss Exponents | 4.0 | Bandwidth | 1.0 GHz |
| Number of microcells | 6 | Data growth ratio | 3 |
| Task size | [1,10] Mbits | Task workload | $[10^2, 10^3]$ cycles/bit |

### A. Weighted-sum system latency

In this subsection, simulations on weighted-sum system latency of five models (i.e., latency-optimal partial offloading model, close-to-optimal partial offloading model, binary offloading model in [7], cloud computing model and local computing model) are demonstrated. Computing capacities of all devices follow the uniform distribution with $V_d \in [0.5 \times 10^9, 4.5 \times 10^9]$. The locations of mobile devices follow the two-dimension uniform distribution. Simulations on system latency versus the number of devices and task workload of five different models are shown in Fig. 3.

Authorized licensed use limited to: Tsinghua University. Downloaded on September 02,2024 at 02:51:18 UTC from IEEE Xplore. Restrictions apply.




(a) Vs. Number of Devices (b) Vs. Task Workload

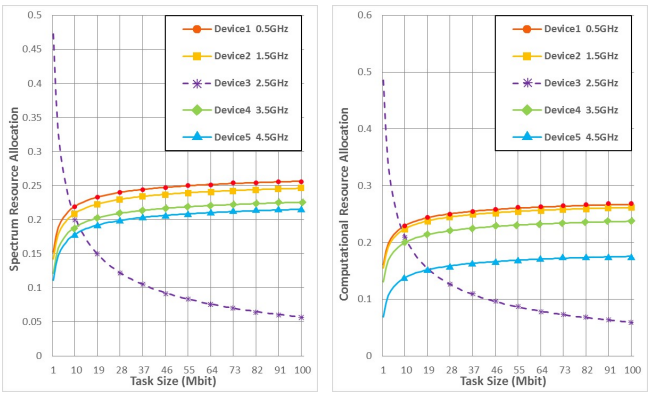
Fig. 3. Average System Latency

Fig. 3(a) shows the latency of five models all increase with the number of devices since computational and radio resources are limited. The derivative of local computing model is the smallest since the radio resource is relatively adequate compared with the computational resource. With the increment of mobile devices, the latency of cloud computing model becomes longer than that of local computing model since computing capacity of MEC server is limited, once the number of devices is high enough, computing capacity allocated to each device would be much smaller than the computing capacity of local device. Two partial offloading models (i.e., optimal and close-to-optimal models) both perform better than binary offloading model since the binary offloading model always chooses a better way (i.e., offload the task or not) while partial offloading model processes every task in parallel. We can see that partial offloading models reduce 10.8% to 39.3% latency compared with binary offloading.

As shown in Fig. 3(b), with the increment of task workload, system latency becomes longer, the performance gaps between our strategies and other strategies become more evident. Compared with cloud computing, partial offloading reduce more than 40% latency and thereby can significantly improve QoE for end-users who require computation-intensive services.

### B. *Resource Allocation*

In this subsection, we analyze the radio and computational resource allocation of tasks requested by five mobile devices with different computing capacities. Note that *Device 3* is a baseline with a constant task size $S_3 = 5$ Mbits.




(a) Radio resource allocation (b) Computational resource allocation

Fig. 4. Resource allocation proportion vs. task size

Fig. 4 illustrates that more resources will be allocated to devices with heavier tasks. Besides, the system prefers to allocate resources to devices with inadequate computing capacities. What's more, the proportion gaps between five devices in Fig. 4(b) are much larger than that in Fig. 4(a) because the limitation of computational resource is the major problem for computation-intensive services.

## VI. Conclusion

In this paper, we investigated the latency optimization of MEC system in 5G heterogeneous networks. We formulated the optimization problem based on partial offloading. After being transformed into a piecewise convex problem, the problem can be solved by sub-gradient method. To reduce the complexity of the strategy, a simplified algorithm based on a common scenario was derived, which has a great potential for practical implementation. The effectiveness of our proposed strategy is verified by numerical results of our experiments.

## References

[1] 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 & Tutorials, vol. PP, no. 99, pp. 1-1.

[2] ETSI, "Mobile Edge Computing, a key technology towards 5G," *White Paper, Mobile-edge Computing Industry Initiative.* [Online]. Available:http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf

[3] G. Orsini, D. Bade and W. Lamersdorf, "Computing at the Mobile Edge: Designing Elastic Android Applications for Computation Offloading," 2015 8th IFIP Wireless and Mobile Networking Conference (WMNC), Munich, 2015, pp. 112-119.

[4] H. Suo, Z. Liu, J. Wan and K. Zhou, "Security and privacy in mobile cloud computing," 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, 2013, pp. 655-659.

[5] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," in IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1628-1656, thirdquarter 2017.

[6] Y. H. Kao et al., "Hermes: Latency Optimal Task Assignment for Resource-constrained Mobile Computing," in IEEE Transactions on Mobile Computing, vol. 16, no. 11, pp. 3056-3069, Nov. 1 2017.

[7] J. Liu et al., "Delay-optimal computation task scheduling for mobile-edge computing systems," 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, 2016, pp. 1451-1455.

[8] 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. PP, no. 99, pp. 1-1.

[9] Y. Wang et al., "Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling," in IEEE Transactions on Communications, vol. 64, no. 10, pp. 4268-4282, Oct. 2016.

[10] C. You et al., "Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading," in IEEE Transactions on Wireless Communications, vol. 16, no. 3, pp. 1397-1411, March 2017.

[11] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in Proc. IEEE Global Commun. Conf., Washington, DC, Dec. 2016, pp. 1-6.

[12] K. Zhang et al., "Energy Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks," in IEEE Access, vol. 4, pp. 5896-5907.

[13] Y. Mao et al., "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.

[14] J. Liu et al., "Content caching at the wireless network edge: A distributed algorithm via belief propagation," 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, 2016, pp. 1-6.

Authorized licensed use limited to: Tsinghua University. Downloaded on September 02,2024 at 02:51:18 UTC from IEEE Xplore. Restrictions apply.