Latency Optimization for Resource Allocation in Mobile-Edge Computation Offloading

Jinke Ren, Guanding Yu, Yunlong Cai, and Yinghui He

***Abstract*—By offloading intensive computation tasks to the edge cloud located at the cellular base stations, mobile-edge computation offloading (MECO) has been regarded as a promising means to accomplish the ambitious millisecond-scale end-to-end latency requirement of the fifth-generation networks. In this paper, we investigate the latency-minimization problem in a multi-user time-division multiple access MECO system with joint communication and computation resource allocation. Three different computation models are studied, i.e., local compression, edge cloud compression, and partial compression offloading. First, closed-form expressions of optimal resource allocation and minimum system delay for both local and edge cloud compression models are derived. Then, for the partial compression offloading model, we formulate a piecewise optimization problem and prove that the optimal data segmentation strategy has a piecewise structure. Based on this result, an optimal joint communication and computation resource allocation algorithm is developed. To gain more insights, we also analyze a specific scenario where communication resource is adequate while computation resource is limited. In this special case, the closed-form solution of the piecewise optimization problem can be derived. Our proposed algorithms are finally verified by numerical results, which show that the novel partial compression offloading model can significantly reduce the end-to-end latency.**

***Index Terms*—Mobile edge computation offloading (MECO), local compression, edge cloud compression, partial compression offloading, resource allocation, piecewise optimization, data segmentation strategy.**

I. Introduction

Over the past few years, the explosive popularity of mobile devices, such as smart-phones, tablets, and wearable devices, has been accelerating the development of the Internet of Things (IoT) [1], [2]. According to the prediction by Cisco, nearly 50 billion IoT devices will be connected to the Internet by 2020, most of which have limited resources for communication, computation, and storage [3]. Due to the exponential growth of mobile data traffic, merely relying on traditional cloud computing is not adequate to realize this ambitious millisecond-scale latency for communication and computation in 5G networks. To keep up with this persistent demand and improve the quality of experience (QoE) for users, the emerging technology of mobile edge computing (MEC) has been gaining significant attention from both academia and industry.

MEC offers application developers and content providers cloud-computing capabilities at the very edge of the mobile network by implementing MEC servers at cellular base stations (BSs), which is also referred to as *edge cloud* [4]. Owing to the close distance from the mobile device to the cloud server, MEC has the potential to fulfill the critical end-to-end delay requirement of 5G networks. Moreover, through mobile edge computation offloading (MECO), the energy consumption of mobile devices can be also reduced by offloading intensive computation workload to the proximate MEC server for execution [5].

The minimization of end-to-end delay and energy consumption in the MECO technique requires joint allocation of communication and computation resources among mobile devices and MEC servers. Recent years have seen lots of studies on this topic for both single-user [6]–[13] and multi-user [14]–[19] MECO systems. In [6] and [7], the authors have derived the optimal resource allocation solution for a single-user MECO system with multiple elastic tasks to minimize the average execution latency of all tasks under the transmit power constraint. On the other hand, to reduce the total energy consumption under a given latency requirement, the authors in [8] have derived the optimal threshold-based offloading policy with joint communication and computation resource allocation and the authors in [9] have proposed the optimal mode selection between local computing and cloud computing. Furthermore, a delay-optimal problem in a single-user MECO system with a prescribed resource utilization constraint has been studied in [10], and a polynomial-time approximate solution with guaranteed performance has been developed herein. Optimal resource allocation and offloading decision policy has been further investigated to minimize the weighted-sum mobile energy consumption under the computation latency constraint [14]. Besides, an online joint communication and computation resource management algorithm for a multi-user MECO system has been developed to minimize the long-term average weighted-sum energy consumption of mobile devices and the cloud server under the buffer stability constraint [15]. A stochastic task arrival model based on the Lyapunov optimization algorithm has been proposed to solve the energy-latency tradeoff problem for a multi-user MECO system [16]. In addition, to minimize the total energy consumption and offloading latency, game-theoretic techniques have been applied to develop the distributed algorithm, which is able to achieve a Nash equilibrium [17]. Moreover, to cope with

Manuscript received August 18, 2017; revised April 10, 2018; accepted June 03, 2018. This work was supported in part by the Natural Science Foundation of China under Grants 61671407 and 61471319, in part by the Open Research Fund of State Key Laboratory of Integrated Services Networks, Xidian University under Grant ISN18-13, and in part by the Zhejiang Provincial Key Project of Research and Development under Grant 2017C01024. The associate editor coordinating the review of this paper and approving it for publication was M. Li. (*Corresponding author: G. Yu.*)

J. Ren and G. Yu are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, and also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xian 710071, China. (e-mail: {renjinke, yuguanding}@zju.edu.cn).

Y. Cai and Y. He are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China. (e-mail: {ylcai, 2014hyh}@zju.edu.cn)

the bursty task arrivals, the MEC server can be integrated with uplink-downlink transmission scheduling to minimize the average latency [18].

Most aforementioned works on multi-user MECO systems focus on the binary computation offloading strategy, i.e., the computation task is executed either at the mobile device or at the edge cloud. Although the pioneering work in [14] has studied the energy-efficient partial computation offloading, the latency-minimization issue has not been discussed therein, which is a more urgent design target for 5G networks. In addition, the authors in [20] have proposed a polynomial-time approximate scheme and an online learning algorithm to minimize the end-to-end latency in a single-task MEC system under a prescribed resource utilization constraint. The latency performance is significantly enhanced as compared against a benchmark heuristic algorithm. This inspiring work mainly focuses on algorithm design while no closed-form results on communication and computation resource allocation have been derived therein. Inspired by this, we investigate the latency-minimization problem in a multi-user MECO system with partial computation offloading in this paper. We assume that mobile devices have a large volume of raw data that are required to be compressed and uploaded to the edge cloud for analysis and storage. This considered scenario is corresponding to the surveillance and security application where massive online monitoring data should be timely transmitted to and analyzed by a central unit. Our design objective is to minimize the weighted-sum delay of all devices under the limited communication and computation resource constraints. According to where the data is compressed, we propose three different models: *local compression* where data is compressed only at mobile devices, *edge cloud compression* where data is transmitted to the edge cloud for compression, and *partial compression offloading* where partial data is compressed locally while the other part is compressed at the edge cloud. The main contributions of this work are summarized as follows.

- For the local compression model, we formulate a convex optimization problem to minimize the weighted-sum delay of all devices under the communication resource constraint. Both closed-form expressions of optimal resource allocation and minimum weighted-sum delay are derived, and some inherent insights are also highlighted.
- For the edge cloud compression model, we analyze the task completion process by modeling a joint resource allocation problem with the constraints of both communication and computation resources. Then the closed-form solution and the minimum weighted-sum delay of all devices can be obtained by utilizing the Lagrange multiplier method.
- For the partial compression offloading model, we first formulate a piecewise optimization problem and then derive the optimal data segmentation strategy in a piecewise structure. Based on this result, we transform the original problem into a piecewise convex problem and develop an optimal resource allocation solution based on the sub-gradient algorithm.
- To yield more insights into the partial compression offloading model, we investigate a common scenario where communication resource is adequate while computation resource is limited. In this specific case, the delay expression of each device can be simplified and the closed-form solution of the piecewise optimization problem can be derived. It is also verified by numerical simulation that the proposed solution for this specific scenario can achieve a near-optimal performance in general scenarios.

The rest of this paper is organized as follows. In Section II, we introduce the multi-user MECO system and the three different compression models. In Section III, we investigate the local compression model and present a closed-form solution to the latency-minimization problem. The resource allocation problem for the edge cloud compression model is analyzed in Section IV. Section V investigates the partial compression offloading model and the closed-form solution for a specific scenario is also analyzed in this section. Simulation results are presented in Section VI and the whole paper is concluded in Section VII.

II. System Model

In this section, we first introduce the multi-user MECO system. After that, we analyze the end-to-end delay of each device in the three models of local compression, edge cloud compression, and partial compression offloading, respectively.

A. Multi-user MECO System

As depicted in Fig. 1, we consider a multi-user MECO system consisting of one edge cloud platform and K single-antenna mobile devices, denoted by a set $\mathcal{K} = \{1, 2, \cdots, K\}$. The edge cloud can be regarded as a data center that is connected by mobile devices through wireless channels. Each device has a volume of data, such as a raw video, that needs to be compressed and stored in the edge cloud.[1] Our system model corresponds to many practical scenarios, such as the surveillance and security monitoring systems where a large number of network cameras have video data to be transmitted to a central unit for further analysis and storage.[2]

The mobile device k in the system can be characterized by two parameters, i.e., the size of raw video $L_k > 0$ (in bits) and the CPU compression capacity V_k^{d} (in bits/s). Denote the total compression capacity of the edge cloud as V^{c} (in bits/s), which can be allocated to all devices. That is, device k will be allocated V_k^{c} computational resource with constraint $\sum_{k=1}^{K} V_k^{\mathrm{c}} \leq V^{\mathrm{c}}$. Additionally, some other reasonable assumptions used in this paper are described as follows.

- The edge cloud has perfect knowledge of the channel gains and the size of videos of all devices, which is

[1] In this paper, we take the video compression as the example for the following analysis whereas our proposed framework can be extended into many data analytical systems, such as image processing and augmented reality, with a minor modification on delay analysis. The closed-form results derived in this paper can be regarded as significant references in other heterogeneous data systems due to the universal data processing model used in our analysis.

[2] Note that in some practical scenarios, to avoid the network congestion and extend the coverage of the systems, the video data must be transmitted through cellular networks instead of other free networks like WiFi. Therefore, monetary costs are not considered in this paper.

required by the centralized scheduler. With this assumption, some insightful results can be obtained in this work, such as the optimal video segmentation strategy and the joint communication and computation resource allocation. Nevertheless, the latency analysis with imperfect channel state information (CSI) and video size information for practical systems deserves further investigation.

- To guarantee that all videos can be compressed at the edge cloud simultaneously, we require that all devices and the edge cloud utilize the same video compression technology, such as MPEG4. Moreover, to model the compression complexity, we define the compression ratio $\beta \in (0,1)$, i.e., the ratio between the compressed data size and the overall raw data size. Note that this definition will be used to model the transmission delay of the local compressed video data later.
- The delay for video segmenting, stitching, and storing can be reasonably neglected since they are much shorter than both communication and computational delays.

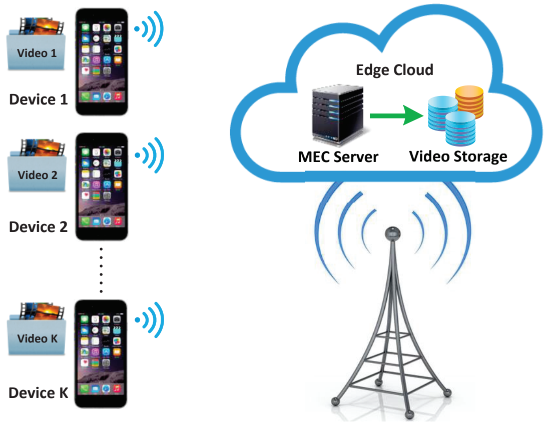


Fig. 1. Multi-user MECO system model.

B. Multiple-Access Model

We apply a time division multiple access (TDMA) method for the channel access.[3] In this method, one time frame is divided into K time slots, which will be allocated to K devices. For convenience, we normalize the duration of time-slot allocated to device k as t_k ($t_k \in [0,1]$). Moreover, to avoid additional signaling overhead and ensure a fair comparison among different compression models, each device can only transmit on its own time-slot, which is allocated by the base station [21]. Note that the length of each time frame is short enough (e.g., 10 ms in LTE standards), which can be reasonably ignored when calculating the end-to-end delay of each device.

Let $h_k(i)$ denote the channel gain of device k in time-slot i, which is a random variable and independently and identically distributed (i.i.d.) across the time-slot. Define p_k as the transmission power of device k. Note that the data compression would incur high computational overhead and thus consume additional power. However, this power is usually fixed and much lower than the transmission power. With this consideration, the transmission power p_k can be determined by subtracting the compression power from the total power budget of each device. Moreover, as indicated in Section I, the focus of our paper is optimizing the delay performance rather than the energy efficiency in [14], [19], [22]. Therefore, the compression power is not emphasized here. Based on the above definition, the achievable data rate in time-slot i can be expressed as

$$r_k(i) = B \log_2 \left(1 + \frac{p_k \left| h_k(i) \right|^2}{N_0} \right), \tag{1}$$

where B and N_0 are the bandwidth and the variance of additive white Gaussian noise (AWGN), respectively.

C. Local Compression Model

In the local compression model, each raw video is compressed locally and then transmitted to the edge cloud for storage. There are two kinds of delay in this model:

- The delay for compressing the raw video L_k bits at device k, $D^{\mathrm{d}}_{\mathrm{comp},k} = \frac{L_k}{V^{\mathrm{d}}_k}$.
- The delay for transmitting the compressed video βL_k bits to the edge cloud, $D^{\mathrm{d}}_{\mathrm{tran},k}$.

It should be emphasized that the compression delay usually has the timescale of second and is not a random variable. On the contrary, the transmission delay is a random variable due to the channel dynamics, which are typically in the timescale of millisecond. From the long timescale perspective, it is reasonable to use the average transmission delay instead of the instantaneous transmission delay. We assume that it takes at least N_k^* time slots for device k to transmit the compressed video k of βL_k bits to the edge cloud, where N_k^* satisfies

$$N_k^* = \arg\min \left\{ N : \sum_{i=1}^{N} r_k(i) \geq \frac{\beta L_k}{T_k} \right\}, \tag{2}$$

where T_k is the duration of time-slot allocated to device k. Note that the transmission rate $r_k(i)$ $(i = 1, 2, \cdots, N)$, as a function of the random channel gain $h_k(i)$, is also a random variable and i.i.d. across time slots. Therefore N_k^* is also a random variable. However, according to [23] and the martingale theory [24], we can evaluate the average transmission delay as

$$D^{\mathrm{d}}_{\mathrm{tran},k} = \mathbb{E}_{\boldsymbol{h}} \left\{ T_k N_k^* \right\} = \frac{\beta L_k}{\mathbb{E}_{\boldsymbol{h}} \left\{ t_k r_k \right\}} = \frac{\beta L_k}{t_k \mathbb{E}_{\boldsymbol{h}} \left\{ r_k \right\}}, \tag{3}$$

where $\mathbb{E}_{\boldsymbol{h}} \left\{ \cdot \right\}$ is the expectation over the channel gain $h_k(i)$.

For ease of notation, we define $R_k = \mathbb{E}_{\boldsymbol{h}} \left\{ r_k \right\}$, which can be regarded as the average data rate of device k across time slots. Moreover, we assume that the transmission can happen only after all data of the raw video is completely compressed. Then the end-to-end delay for device k to complete its task

[3] Although we assume a TDMA scenario, our analysis can be also extended into other orthogonal and non-orthogonal multiple access schemes for 5G networks with a minor modification on the delay model.

can be expressed as

$$D_k = D^{\mathrm{d}}_{\mathrm{comp},k} + D^{\mathrm{d}}_{\mathrm{tran},k} = \frac{L_k}{V^{\mathrm{d}}_k} + \frac{\beta L_k}{t_k R_k}. \tag{4}$$

D. Edge Cloud Compression Model

In the edge cloud compression model, each device directly uploads its raw video to the edge cloud without any compression. Thereafter, the edge cloud compresses all raw videos in parallel by optimally allocating its computation resource. Similar to the local compression model, there also exist two kinds of delay in this model:

- The delay for transmitting the raw video L_k bits to the edge cloud, $D^{\mathrm{c}}_{\mathrm{tran},k} = \dfrac{L_k}{t_k R_k}$.
- The delay for compressing the raw video L_k bits at the edge cloud, $D^{\mathrm{c}}_{\mathrm{comp},k} = \dfrac{L_k}{V^{\mathrm{c}}_k}$.

Correspondingly, we require that the edge cloud can start compressing a raw video only after completely receiving its whole data. Then the end-to-end delay for device k to complete its task can be written as

$$D_k = D^{\mathrm{c}}_{\mathrm{tran},k} + D^{\mathrm{c}}_{\mathrm{comp},k} = \frac{L_k}{t_k R_k} + \frac{L_k}{V^{\mathrm{c}}_k}. \tag{5}$$

E. Partial Compression Offloading Model

In the local compression model, the local compression delay, $D^{\mathrm{d}}_{\mathrm{comp},k}$, would be dominant if the speed of device CPU is limited, i.e., corresponding to the wireless video monitoring camera. On the other hand, in the edge cloud compression model, the transmission delay, $D^{\mathrm{c}}_{\mathrm{tran},k}$, would be dominant if the channel bandwidth is limited. Obviously, both two models are not optimal in terms of end-to-end delay minimization if the video can be partially compressed at the mobile device and partially compressed at the edge cloud. Motivated by this, in this subsection, we propose a partial compression offloading model, in which each raw video can be partitioned into two parts with one compressed locally while the other offloaded for edge compression. Let us denote the proportion of video k that is compressed at the mobile device as $\lambda_k \in [0, 1]$. Then, we can introduce the detailed procedure of partial compression offloading in three steps, as depicted in Fig. 2.

- Mobile device k compresses $\lambda_k L_k$ bits of the raw video locally and then transmits the compressed data of $\beta \lambda_k L_k$ bits to the edge cloud.
- Mobile device k transmits the remaining $(1 - \lambda_k)L_k$ bits to the edge cloud. Then the edge cloud compresses this part of raw video itself.
- Finally, the edge cloud combines two parts of compressed video and stores them into the data-storage center.

In the above three steps, there exist four kinds of delay as follows.

- The delay for compressing $\lambda_k L_k$ bits at mobile device k, $D^{\mathrm{d}}_{\mathrm{comp},k} = \dfrac{\lambda_k L_k}{V^{\mathrm{d}}_k}$.
- The delay for transmitting the local compressed part, $D^{\mathrm{d}}_{\mathrm{tran},k} = \dfrac{\beta \lambda_k L_k}{t_k R_k}$.

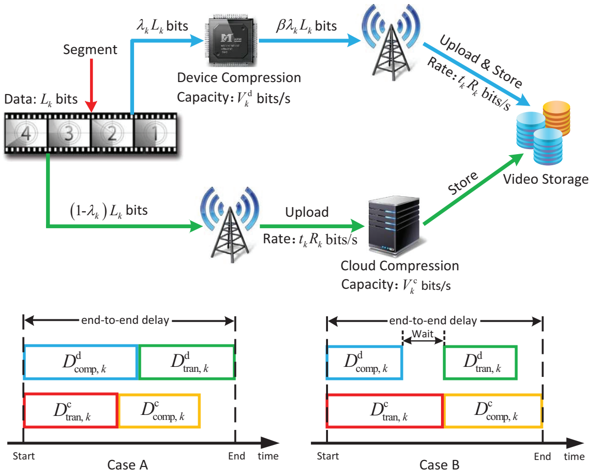

Segment
$\lambda_k L_k$ bits
$\beta\lambda_k L_k$ bits
Data: L_k bits
4
3
2
1
Device Compression
Capacity: V^{d}_k bits/s
Upload & Store
Rate: $t_k R_k$ bits/s
Video Storage
Store
$(1\text{-}\lambda_k)\,L_k$ bits
Upload
Rate: $t_k R_k$ bits/s
Cloud Compression
Capacity: V^{c}_k bits/s
end-to-end delay
$D^{\mathrm{d}}_{\mathrm{comp},\,k}$
$D^{\mathrm{d}}_{\mathrm{tran},\,k}$
$D^{\mathrm{c}}_{\mathrm{tran},\,k}$
$D^{\mathrm{c}}_{\mathrm{comp},\,k}$
Start
End
time
Case A
end-to-end delay
Wait
$D^{\mathrm{d}}_{\mathrm{comp},\,k}$
$D^{\mathrm{d}}_{\mathrm{tran},\,k}$
$D^{\mathrm{c}}_{\mathrm{tran},\,k}$
$D^{\mathrm{c}}_{\mathrm{comp},\,k}$
Start
End
time
Case B

Fig. 2. The whole process of partial compression offloading.

- The delay for transmitting the uncompressed part, $D^{\mathrm{c}}_{\mathrm{tran},k} = \dfrac{(1 - \lambda_k)L_k}{t_k R_k}$.
- The delay for compressing $(1 - \lambda_k)L_k$ bits at the edge cloud, $D^{\mathrm{c}}_{\mathrm{comp},k} = \dfrac{(1 - \lambda_k)L_k}{V^{\mathrm{c}}_k}$.

Since each device has only one channel for data transmission, either local compression part or edge cloud compression part can be transmitted at any moment while not simultaneously. Therefore, as depicted in Fig. 2, two cases will happen. The first one corresponds to $D^{\mathrm{d}}_{\mathrm{comp},k} \geq D^{\mathrm{c}}_{\mathrm{tran},k}$, where the transmission for local compressed video can start immediately at the end of the local compression. The second one corresponds to $D^{\mathrm{d}}_{\mathrm{comp},k} < D^{\mathrm{c}}_{\mathrm{tran},k}$, where the transmission for local compressed video must wait until the transmission for the edge cloud compression part ends. Therefore, the end-to-end delay of device k in this model can be written as

$$D_k = \begin{cases} \max\left\{D^{\mathrm{d}}_{\mathrm{comp},k} + D^{\mathrm{d}}_{\mathrm{tran},k}, D^{\mathrm{c}}_{\mathrm{tran},k} + D^{\mathrm{c}}_{\mathrm{comp},k}\right\}, \\ \qquad\qquad\qquad\qquad \text{if } D^{\mathrm{d}}_{\mathrm{comp},k} \geq D^{\mathrm{c}}_{\mathrm{tran},k}, \\ D^{\mathrm{c}}_{\mathrm{tran},k} + \max\left\{D^{\mathrm{d}}_{\mathrm{tran},k}, D^{\mathrm{c}}_{\mathrm{comp},k}\right\}, \\ \qquad\qquad\qquad\qquad \text{if } D^{\mathrm{d}}_{\mathrm{comp},k} < D^{\mathrm{c}}_{\mathrm{tran},k}. \end{cases} \tag{6}$$

In the next three sections, we will develop optimal joint communication and computation resource allocation algorithms to minimize the weighted-sum delay of all devices for the three different models, respectively.

III. Optimal Solution to the Local Compression Model

In this section, we first formulate the latency-minimization problem for the local compression model and then derive the closed-form expressions for both optimal solution and minimum weighted-sum delay of all devices.

A. Problem Formulation

We aim at minimizing the weighted-sum end-to-end delay of all tasks, $\sum_{k=1}^{K} \alpha_k D_k$, where the positive weight factors $\{\alpha_k\}$ account for the fairness among devices and satisfy

$\sum_{k=1}^{K} \alpha_k = 1$. The end-to-end delay of a task may include the delay of local compression, the delay of data transmission, and the delay of edge compression. Therefore, the objective function reflects the average delay performance of the whole system, which is essential to improve the QoE of all devices. Based on the end-to-end delay expression in (4), we have the following optimization problem for the local compression model.

Problem 1: (Local Compression)

$$\min_{\{t_k\}} \quad \sum_{k=1}^{K} \alpha_k \left(\frac{L_k}{V_k^{\mathrm{d}}} + \frac{\beta L_k}{t_k R_k} \right), \tag{7a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} t_k \le 1, \ t_k \ge 0, \tag{7b}$$

where (7b) is the overall communication resource constraint of all devices.

B. Optimal Solution

It can be easily verified that Problem 1 is convex and the Slater's condition can be satisfied, implying that strong duality holds. Therefore, Problem 1 can be solved by the Karush-Kuhn-Tucker (KKT) conditions and we can derive the following optimal solution, as presented in Theorem 1.

Theorem 1: The optimal solution solving Problem 1 of the local compression model is given by

$$t_k^{*(1)} = \frac{\sqrt{\dfrac{\alpha_k L_k}{R_k}}}{\sum_{k=1}^{K} \sqrt{\dfrac{\alpha_k L_k}{R_k}}}, \ \forall k \in \mathcal{K}. \tag{8}$$

Remark 1: Theorem 1 reveals that the optimal time-slot allocated to device k is determined by the corresponding weight factor, size of raw video, and channel capacity. The weight factor, α_k, can be interpreted as the level of importance for device k. The larger the value of α_k is, the more the time-slot should be allocated to device k to minimize the whole system delay. Furthermore, more time-slot should be allocated to device k if the video size, L_k, becomes larger or the channel capacity, R_k, gets smaller.

Based on the above solution, we can derive the minimum system delay (i.e., weighted-sum delay of all devices) in a closed-form way, as

$$\begin{aligned} D_{\text{sys}}^{\mathrm{L}} &= \sum_{k=1}^{K} \frac{\alpha_k L_k}{V_k^{\mathrm{d}}} + \sum_{i=1}^{K} \left(\sqrt{\alpha_i \frac{\beta L_i}{R_i}} \sum_{j=1}^{K} \sqrt{\alpha_j \frac{\beta L_j}{R_j}} \right) \\ &= \sum_{k=1}^{K} \frac{\alpha_k L_k}{V_k^{\mathrm{d}}} + \beta \sum_{i=1}^{K} \sum_{j=1}^{K} \sqrt{\frac{\alpha_i \alpha_j L_i L_j}{R_i R_j}}. \end{aligned} \tag{9}$$

IV. Optimal Solution to the Edge Cloud Compression Model

In this section, we analyze the latency-minimization problem for the edge cloud compression model and devise the joint optimal communication and computation resource allocation algorithm.

A. Problem Formulation

To minimize the weighted-sum delay of all devices, we have the following problem for the edge cloud compression model.

Problem 2: (Edge Cloud Compression)

$$\min_{\{t_k, V_k^{\mathrm{c}}\}} \quad \sum_{k=1}^{K} \alpha_k \left(\frac{L_k}{t_k R_k} + \frac{L_k}{V_k^{\mathrm{c}}} \right), \tag{10a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} t_k \le 1, \ t_k \ge 0, \tag{10b}$$

$$\sum_{k=1}^{K} V_k^{\mathrm{c}} \le V^{\mathrm{c}}, \ V_k^{\mathrm{c}} \ge 0, \tag{10c}$$

where constraints (10b) and (10c) imply that the overall communication and computation resources allocated to mobile devices cannot exceed the corresponding limitations.

B. Optimal Solution

Fortunately, Problem 2 is also convex since each component in (10a) is convex on t_k and V_k^{c}. Therefore, Problem 2 can be optimally solved using the KKT conditions and we can obtain the optimal solution, as shown in Theorem 2.

Theorem 2: The optimal solution for Problem 2 of the edge cloud compression model is given by

$$\begin{cases} t_k^{*(2)} = \dfrac{\sqrt{\dfrac{\alpha_k L_k}{R_k}}}{\sum_{k=1}^{K} \sqrt{\dfrac{\alpha_k L_k}{R_k}}}, \ \forall k \in \mathcal{K}, \\ V_k^{\mathrm{c}*(2)} = \dfrac{\sqrt{\alpha_k L_k}}{\sum_{k=1}^{K} \sqrt{\alpha_k L_k}} V^{\mathrm{c}}, \ \forall k \in \mathcal{K}. \end{cases} \tag{11}$$

Remark 2: From Theorem 2, we can see that the optimal time-slot allocated to each device in the edge cloud compression model has the same expression as that in the local compression model, and the optimal cloud compression capacity allocated to each device is determined by the corresponding weight factor and video size. Similarly, the weight factor α_k is positively related to the allocated resources since it reflects the level of importance for device k. Specially, in case that each device has the same weight, i.e., $\alpha_k = \dfrac{1}{K}, \forall k \in \mathcal{K}$, the bigger the video size is, the more the time-slot and edge cloud compression capacity should be allocated to the device for achieving the minimum system delay. Again, we can express the minimum system delay in a closed-form way, as

$$\begin{aligned} & D_{\text{sys}}^{\mathrm{E}} \\ &= \sum_{i=1}^{K} \left(\sqrt{\alpha_i \frac{L_i}{R_i}} \sum_{j=1}^{K} \sqrt{\alpha_j \frac{L_j}{R_j}} \right) + \sum_{i=1}^{K} \left(\sqrt{\alpha_i L_i} \sum_{j=1}^{K} \frac{\sqrt{\alpha_j L_j}}{V^{\mathrm{c}}} \right) \\ &= \sum_{i=1}^{K} \sum_{j=1}^{K} \sqrt{\alpha_i \alpha_j L_i L_j} \left(\sqrt{\frac{1}{R_i R_j}} + \frac{1}{V^{\mathrm{c}}} \right). \end{aligned} \tag{12}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2018.2845360, IEEE Transactions on Wireless Communications

V. Optimal Solution to the Partial Compression Offloading Model

In the above sections, we have analyzed the optimal communication and computation resource allocations for the local compression model and the edge cloud compression model, respectively. In this section, we shall investigate the latency-minimization problem for the partial compression offloading model. The problem studied in this section is more general in that it fully utilizes the computation resource in both mobile devices and the edge cloud, which can further reduce the system delay. In the following, we shall first formulate the latency-minimization problem and then derive the optimal video segmentation strategy in a piecewise structure. After that, we will transform the original problem into a piecewise convex problem and develop a sub-gradient algorithm to find the optimal solution efficiently. Finally, the closed-form solution in a specific scenario will be also devised.

A. Problem Formulation

In the partial compression offloading model, each raw video could be partially compressed at the mobile device and partially compressed at the edge cloud. Therefore, both communication and computation resources should be jointly allocated and the optimization problem can be formulated as

Problem 3: (Partial Compression Offloading)

$$\min_{\{t_k, V_k^{\mathrm{c}}, \lambda_k\}} \quad \sum_{k=1}^{K} \alpha_k D_k, \tag{13a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} t_k \le 1, \ t_k \ge 0, \tag{13b}$$

$$\sum_{k=1}^{K} V_k^{\mathrm{c}} \le V^{\mathrm{c}}, \ V_k^{\mathrm{c}} \ge 0, \tag{13c}$$

$$0 \le \lambda_k \le 1, \ \forall k \in \mathcal{K}. \tag{13d}$$

Notice that D_k is a piecewise function given in (6), which can be rewritten in a more detailed way, as

$$D_k = \begin{cases} \max\left\{ \dfrac{\lambda_k L_k}{V_k^{\mathrm{d}}} + \dfrac{\beta \lambda_k L_k}{t_k R_k}, \dfrac{(1-\lambda_k) L_k}{t_k R_k} + \dfrac{(1-\lambda_k) L_k}{V_k^{\mathrm{c}}} \right\}, \\ \qquad \text{if } \lambda_k \ge \dfrac{V_k^{\mathrm{d}}}{V_k^{\mathrm{d}} + t_k R_k}, \\ \dfrac{(1-\lambda_k) L_k}{t_k R_k} + \max\left\{ \dfrac{\beta \lambda_k L_k}{t_k R_k}, \dfrac{(1-\lambda_k) L_k}{V_k^{\mathrm{c}}} \right\}, \\ \qquad \text{if } \lambda_k < \dfrac{V_k^{\mathrm{d}}}{V_k^{\mathrm{d}} + t_k R_k}. \end{cases} \tag{14}$$

B. Optimal Segmentation Strategy and Problem Transformation

It can be seen that the D_k expression in (14) is complicated with $3K$ variables such that Problem 3 is hard to be solved directly. In the following, we will determine the optimal λ_k^* while keeping t_k and V_k^{c} fixed. First, let us define a geometric mean $\sqrt{\beta V_k^{\mathrm{d}} V_k^{\mathrm{c}}}$, which is referred to as the *average compression capacity* for compressing video k. Correspondingly, $t_k R_k$ can be regarded as the *average communication capacity* for transmitting video k. Then the optimal video segmentation strategy has the following piecewise structure, as presented in Lemma 1.

Lemma 1: Given the sets of $\{t_k\}$ and $\{V_k^{\mathrm{c}}\}$, the optimal video segmentation strategy for each device is given by

$$\lambda_k^* = \begin{cases} \dfrac{V_k^{\mathrm{d}} \left(t_k R_k + V_k^{\mathrm{c}} \right)}{V_k^{\mathrm{d}} V_k^{\mathrm{c}} (1+\beta) + t_k R_k (V_k^{\mathrm{d}} + V_k^{\mathrm{c}})}, & \text{if } t_k R_k \ge \sqrt{\beta V_k^{\mathrm{d}} V_k^{\mathrm{c}}}, \\ \dfrac{V_k^{\mathrm{d}}}{V_k^{\mathrm{d}} + t_k R_k}, & \text{if } t_k R_k < \sqrt{\beta V_k^{\mathrm{d}} V_k^{\mathrm{c}}}. \end{cases} \tag{15}$$

Proof: Please refer to Appendix A. ■

Remark 3: The optimal video segmentation strategy shown in Lemma 1 is determined by comparing the *average communication capacity* with the *average compression capacity* of each device. In the case that the *average communication capacity* dominates the *average compression capacity*, i.e., $t_k R_k \ge \sqrt{\beta V_k^{\mathrm{d}} V_k^{\mathrm{c}}}$, the computation resource is the bottleneck of delay minimization for device k, and therefore we should make full use of the computation resource. In this case, λ_k^* satisfies $D_{\text{comp},k}^{\mathrm{d}} \ge D_{\text{tran},k}^{\mathrm{c}}$ while $D_{\text{comp},k}^{\mathrm{d}} + D_{\text{tran},k}^{\mathrm{d}} = D_{\text{tran},k}^{\mathrm{c}} + D_{\text{comp},k}^{\mathrm{c}}$. On the contrary, in the case that $t_k R_k < \sqrt{\beta V_k^{\mathrm{d}} V_k^{\mathrm{c}}}$, the communication resource is the main bottleneck of delay minimization. Therefore, we need to fully utilize the communication resource to minimize the end-to-end delay of each device. In this case, λ_k^* fulfills that $D_{\text{comp},k}^{\mathrm{d}} = D_{\text{tran},k}^{\mathrm{c}}$ while $D_{\text{comp},k}^{\mathrm{c}} < D_{\text{tran},k}^{\mathrm{d}}$.

By substituting the optimal video segmentation strategy into (14), the end-to-end delay of device k can be written as

$$\widehat{D}_k = \begin{cases} \dfrac{L_k}{t_k R_k} \dfrac{\left(t_k R_k + V_k^{\mathrm{c}}\right)\left(t_k R_k + \beta V_k^{\mathrm{d}}\right)}{V_k^{\mathrm{d}} V_k^{\mathrm{c}} (1+\beta) + t_k R_k (V_k^{\mathrm{d}} + V_k^{\mathrm{c}})} \triangleq \widehat{D}_{k,1}, \\ \qquad \text{if } t_k R_k \ge \sqrt{\beta V_k^{\mathrm{d}} V_k^{\mathrm{c}}}, \\ \dfrac{L_k}{t_k R_k} \dfrac{t_k R_k + \beta V_k^{\mathrm{d}}}{V_k^{\mathrm{d}} + t_k R_k} \triangleq \widehat{D}_{k,2}, \\ \qquad \text{if } t_k R_k < \sqrt{\beta V_k^{\mathrm{d}} V_k^{\mathrm{c}}}. \end{cases} \tag{16}$$

Then Problem 3 can be equivalently converted to the following problem.

Problem 4: (Equivalent Problem of Problem 3).

$$\min_{\{t_k, V_k^{\mathrm{c}}\}} \quad \sum_{k=1}^{K} \alpha_k \widehat{D}_k, \tag{17a}$$

$$\text{s.t.} \quad \text{(13b) and (13c).} \tag{17b}$$

Theorem 3: Problem 4 is a piecewise convex optimization problem.

Proof: Please refer to Appendix B. ■

C. Optimal Resource Allocation Algorithm

The key challenge of Problem 4 is that the $\widehat{D}_k$ expression in (16) is continuous but non-differential (or non-smooth) at $t_k R_k = \sqrt{\beta V_k^{\mathrm{d}} V_k^{\mathrm{c}}}$. Moreover, the partial derivatives of both $\widehat{D}_{k,1}$ and $\widehat{D}_{k,2}$ on t_k have quartic forms. Therefore, classical KKT conditions cannot be directly applied to solve this problem and it is rather difficult to find its closed-form solution. In the following, we will develop an effective algorithm to optimally solve it, which is based on the sub-gradient method for common non-differential convex problems [25]. For ease of notation, we shall first define the following auxiliary variables.

- Define a vector of independent resource variables, as $\boldsymbol{x} = [t_1, t_2, \cdots, t_K, V_1^{\rm c}, V_2^{\rm c}, \cdots, V_K^{\rm c}]$.
- Define the weighted-sum delay of all devices, as $F = \sum_{k=1}^{K} \alpha_k \widehat{D}_k$.

After that, we denote the sub-gradient function of $\widehat{D}_k$ as $\partial \widehat{D}_k = \left[\frac{\partial \widehat{D}_k}{\partial t_k}, \frac{\partial \widehat{D}_k}{\partial V_k^{\rm c}}\right]$. Since it has been proved in Theorem 3 that $\widehat{D}_k$ can be written as $\max\left\{\widehat{D}_{k,1}, \widehat{D}_{k,2}\right\}$, the sub-gradient function can be characterized as

$$\frac{\partial \widehat{D}_k}{\partial t_k} \in \begin{cases} \frac{\partial \widehat{D}_{k,1}}{\partial t_k}, & \text{if } t_k R_k > \sqrt{\beta V_k^{\rm d} V_k^{\rm c}}, \\ \left[\frac{\partial \widehat{D}_{k,2}}{\partial t_k}, \frac{\partial \widehat{D}_{k,1}}{\partial t_k}\right], & \text{if } t_k R_k = \sqrt{\beta V_k^{\rm d} V_k^{\rm c}}, \\ \frac{\partial \widehat{D}_{k,2}}{\partial t_k}, & \text{if } t_k R_k < \sqrt{\beta V_k^{\rm d} V_k^{\rm c}}, \end{cases} \tag{18}$$

$$\frac{\partial \widehat{D}_k}{\partial V_k^{\rm c}} \in \begin{cases} \frac{\partial \widehat{D}_{k,1}}{\partial V_k^{\rm c}}, & \text{if } t_k R_k > \sqrt{\beta V_k^{\rm d} V_k^{\rm c}}, \\ \left[\frac{\partial \widehat{D}_{k,1}}{\partial V_k^{\rm c}}, \frac{\partial \widehat{D}_{k,2}}{\partial V_k^{\rm c}}\right], & \text{if } t_k R_k = \sqrt{\beta V_k^{\rm d} V_k^{\rm c}}, \\ \frac{\partial \widehat{D}_{k,2}}{\partial V_k^{\rm c}}, & \text{if } t_k R_k < \sqrt{\beta V_k^{\rm d} V_k^{\rm c}}. \end{cases} \tag{19}$$

Based on the above analysis, we introduce the following theorem to solve Problem 4.

Theorem 4: Problem 4 can be solved by the following iteration

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \phi_n \boldsymbol{g}^{(n)}, \tag{20}$$

where ϕ_n is the step size of the $n^{\rm th}$ iteration and $\boldsymbol{g}$ is the sub-gradient function of $\sum_{k=1}^{K} \alpha_k \widehat{D}_k$, which is defined as

$$\boldsymbol{g} = \begin{cases} \partial\left(\sum_{k=1}^{K} \alpha_k \widehat{D}_k\right), & \text{subject to (13b) and (13c)}, \\ \partial\left(\sum_{k=1}^{K} t_k\right), & \text{if } \sum_{k=1}^{K} t_k > 1, \\ \partial\left(\sum_{k=1}^{K} V_k^{\rm c}\right), & \text{if } \sum_{k=1}^{K} V_k^{\rm c} > V^{\rm c}, \end{cases} \tag{21}$$

where $\partial\left(\sum_{k=1}^{K} t_k\right)$ and $\partial\left(\sum_{k=1}^{K} V_k^{\rm c}\right)$ are utilized as the obstacle functions.

Proof: Please refer to Appendix C. ■

Based on Theorem 4, we can efficiently solve Problem 4 by iteratively updating the communication and computation resource allocation, whose detailed procedures are presented in Table I.

Now we discuss the convergence and the computational complexity of Algorithm 1. As we have proved in Appendix C, the vector $\boldsymbol{x}^{(n)}$ will linearly converge to the optimal solution $\boldsymbol{x}^*$ when $\epsilon \to 0$ [25], [26]. On the other hand, the computational complexity of the sub-gradient algorithm mainly lies on the required number of iterations until convergence, which is determined by the maximum tolerance ϵ. According to the proof in Appendix C, we can conclude that the required number of iterations of Algorithm 1 is linearly proportional with ϵ^2. Therefore, the proposed algorithm has a polynomial time complexity of $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$, which is desirable for practical implementation.

TABLE I
THE SUB-GRADIENT ALGORITHM FOR THE PARTIAL COMPRESSION OFFLOADING MODEL

Algorithm 1 The sub-gradient algorithm for the partial compression offloading model

```
Initialize
    Initialize the maximum convergence tolerance $\epsilon > 0$.
    Set the iteration index $n = 0$.
    Set the initial resource allocation vector $\boldsymbol{x}^{(0)}$ that subjects to (13b) and (13c).
    Calculate $F^{(0)} = \sum_{k=1}^{K} \alpha_k \widehat{D}_k^{(0)}$ and $\boldsymbol{g}^{(0)}$ according to (16) and (21).
Do
    Update the resource allocation vector by
        $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} - \phi_n \boldsymbol{g}^{(n)}$.
    Update $n = n + 1$.
    Calculate $F^{(n)} = \sum_{k=1}^{K} \alpha_k \widehat{D}_k^{(n)}$ and $\boldsymbol{g}^{(n)}$ according to (16) and (21).
Until $|F^{(n)} - F^{(n-1)}| \le \epsilon$.
```

D. Special Cases

The optimal solution developed in the above subsection is not in closed-form. To yield more insights into the partial compression offloading model, we further investigate a common scenario where the communication resource is adequate while the computation resource is limited, such as the typical sensor network or the machine-type communications. The key characteristic of this specific scenario is that the channel capacity is much greater than the device computation capacity, i.e., $R_k \gg V_k^{\rm d}$. Moreover, most current mobile devices utilize the MPEG4 video compression technology whose compression ratio, β, is between $\frac{1}{50}$ and $\frac{1}{200}$ [27], [28], resulting in the small size of local compressed video. Under these conditions, the delay for transmitting the local compressed video, $D_{{\rm tran},k}^{\rm d}$, can be neglected while comparing with the delay for compressing the local compression part of video, $D_{{\rm comp},k}^{\rm d}$. That is, $\frac{\beta \lambda_k L_k}{t_k R_k} \ll \frac{\lambda_k L_k}{V_k^{\rm d}}$. Therefore, it is straightforward that the optimal video segmentation strategy in this case satisfies

$$D_{{\rm comp},k}^{\rm d} = D_{{\rm tran},k}^{\rm c} + D_{{\rm comp},k}^{\rm c}. \tag{22}$$

Then applying the detailed delay expressions in Section II-E into (22), we have the following optimal video segmentation strategy.

Lemma 2: In the specific scenario of partial compression offloading, the optimal video segmentation strategy for each device is given by

$$\overline{\lambda}_k^* = \frac{V_k^{\mathrm{d}} \left(t_k R_k + V_k^{\mathrm{c}} \right)}{V_k^{\mathrm{d}} V_k^{\mathrm{c}} + t_k R_k \left(V_k^{\mathrm{d}} + V_k^{\mathrm{c}} \right)}, \ \forall k \in \mathcal{K}. \tag{23}$$

Based on Lemma 2, the end-to-end delay of device k can be written as

$$\overline{D}_k = \frac{L_k \left(t_k R_k + V_k^{\mathrm{c}} \right)}{V_k^{\mathrm{d}} V_k^{\mathrm{c}} + t_k R_k \left(V_k^{\mathrm{d}} + V_k^{\mathrm{c}} \right)}. \tag{24}$$

Substituting $\overline{D}_k$ into (17a), the convex Problem 4 can be solved by the KKT conditions. Therefore, the optimal solution for this specific scenario can be derived, as shown in Theorem 5.

Theorem 5: The optimal solution for the specific scenario of partial compression offloading is given by

$$\begin{cases} \overline{t}_k^* = \dfrac{\overline{V}_k^{\mathrm{c}*} \left(\sqrt{\dfrac{\alpha_k L_k R_k}{\theta^*}} - V_k^{\mathrm{d}} \right)^+}{R_k \left(V_k^{\mathrm{d}} + \overline{V}_k^{\mathrm{c}*} \right)}, \ \forall k \in \mathcal{K}, \\ \overline{V}_k^{\mathrm{c}*} = \dfrac{\overline{t}_k^* R_k \left(\sqrt{\dfrac{\alpha_k L_k}{\omega^*}} - V_k^{\mathrm{d}} \right)^+}{\overline{t}_k^* R_k + V_k^{\mathrm{d}}}, \ \forall k \in \mathcal{K}, \end{cases} \tag{25}$$

where $(y)^+ = \max\{y, 0\}$, θ^* and ω^* are the optimal value of Lagrange multipliers that satisfy the active communication and computation resource constraints $\sum_{k=1}^K \overline{t}_k^* = 1$ and $\sum_{k=1}^K \overline{V}_k^{\mathrm{c}*} = V^{\mathrm{c}}$, respectively.

Proof: Please refer to Appendix D. ■

Remark 4: Theorem 5 reveals that the optimal time-slot and cloud compression capacity allocated to device k is determined by the corresponding weight factor, size of raw video, channel capacity, and local compression capacity. As the video size L_k increases, more communication and computation resources will be allocated to this device. However, less communication resource will be allocated if the communication capacity becomes larger. This result is consistent with the intuition that, to reduce the weighted-sum delay of all devices, the BS should allocate more communication resource to those devices with bad channels. Moreover, in case that the local compression capacity V_k^{d} gets smaller, more communication and computation resources should be allocated to this device under the criterion of minimizing the system delay.

For another special case where the computation resource is adequate while the communication resource is deficient, i.e., $V_k^{\mathrm{c}} \gg R_k$, $V_k^{\mathrm{d}} \gg R_k$, the edge cloud compression delay, $D_{\mathrm{comp},k}^{\mathrm{c}}$, can be ignored as compared with the delay for transmitting the cloud compression part of video, $D_{\mathrm{tran},k}^{\mathrm{c}}$. That is, $\dfrac{(1-\lambda_k)L_k}{t_k R_k} \gg \dfrac{(1-\lambda_k)L_k}{V_k^{\mathrm{c}}}$. Therefore, the optimal video segmentation strategy in this case satisfies

$$D_{\mathrm{comp},k}^{\mathrm{d}} = D_{\mathrm{tran},k}^{\mathrm{c}}. \tag{26}$$

By applying the detailed delay expressions into (26), we can derive the optimal video segmentation strategy as

$$\overline{\lambda}_k^* = \frac{1}{1 + \frac{t_k R_k}{V_k^{\mathrm{d}}}} \to 1, \ \forall k \in \mathcal{K}. \tag{27}$$

Based on the above result, we can observe that when the communication resource is the main bottleneck, each video should be compressed locally without offloading to the edge cloud. The reason can be explained as follows. Since the communication resource is deficient, it is better to process the data locally rather than offloading to the edge cloud to avoid large transmission delay. In this case, the optimal communication resource allocation is just the same as that in the local compression model.

VI. Numerical Results

In this section, we will present numerical results to verify our analysis and validate the performance of the proposed algorithms. The simulation settings are as follows unless otherwise stated. The BS has a radius of 250 m. Each mobile device is randomly located in the system and can associate with the BS through one wireless channel. The weights for all devices are the same, i.e., $\alpha_k = \dfrac{1}{K}$ for all k such that the system delay represents the average end-to-end delay of all devices. The channel gains between mobile devices and the edge cloud are generated according to i.i.d. Rayleigh random variables with unit variances. The transmission power is set equal for each device, i.e., $p_k = 24$ dBm, $\forall k \in \mathcal{K}$. The total bandwidth B =10 MHz. For each compression task, the video size and the device compression capacity follow the uniform distribution with $L_k \in$ [10, 100] Mbits and $V_k^{\mathrm{d}} \in$ [0.5, 2] Mbps, respectively. All random variables are independent for different devices, modeling heterogeneous mobile compression capacity. The total compression capacity of the edge cloud V^{c} is selected as 40 Mbps. The compression ratio β is set as 0.01, which is a typical value for the MPEG4 video compression technique. Note that the trends and hierarchies of the curves in the following figures will not change with different values of β. Other major simulation parameters are listed in Table II.

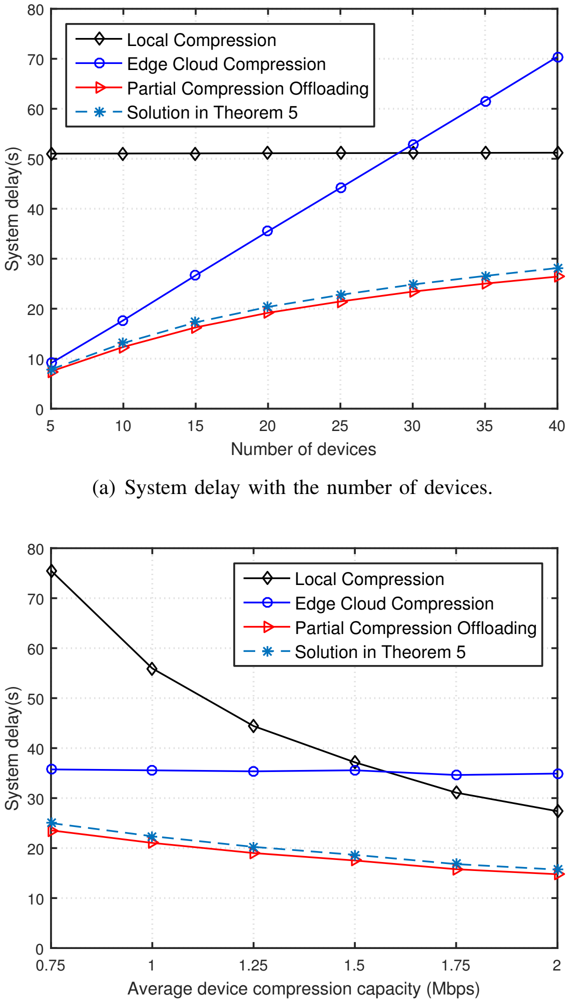
TABLE II
Simulation Parameters

Parameter	Value
Cell radius	250 m
Bandwidth, B	10 MHz
Noise power density	-174 dBm/Hz
Path loss exp.	4
Transmission power, p_k	24 dBm
Raw video size, L_k	[10, 100] Mbits
Device compression capacity, V_k^{d}	[0.5, 2] Mbps
Edge cloud compression capacity, V^{c}	40 Mbps

A. Performance Comparison among Three Models

We first compare the minimum system delays of local compression, edge cloud compression, and partial compression offloading.

Fig. 3(a) depicts the minimum system delay versus the number of mobile devices in the three different models. First, the system delays of the edge cloud compression and partial compression offloading models increase with the number of mobile devices due to the limited computation resource,



(a) System delay with the number of devices.

(b) System delay with the device compression capacity.

Fig. 3. System delay of three models.

while the system delay of the local compression model is approximately invariant since the communication resource is relatively adequate in our simulation. Secondly, by comparing the curves of local compression and edge cloud compression, we can observe that the edge cloud compression performs better than the local compression only when the number of devices is small. The reason can be explained as follows. In case that the number of devices is small, the cloud compression capacity allocated to each device would be larger than the local compression capacity. In this case, it is better to offload computation workload to the edge cloud for compression than local compression from the perspective of delay minimization. On the other hand, as the number of devices grows, the cloud compression capacity allocated to each device would be smaller than the local compression capacity, leading to the better performance of local compression model. Thirdly, the partial compression offloading model has the best performance among the three models since it jointly utilizes the communication and computation resources. The performance gap between the edge cloud compression and partial compression offloading models becomes more evident with the growing number of devices, which indicates that when the number of users becomes large, using the partial compression offloading model can greatly reduce the system delay and improve the QoE for users. Finally, the closed-form solution in Theorem 5 can achieve a near-optimal performance and outperforms both the local compression and edge cloud compression models. It is because under our simulation settings, the compression capacity of mobile devices is much smaller than the corresponding communication capacity. This result demonstrates the effectiveness of our derivation in Theorem 5.

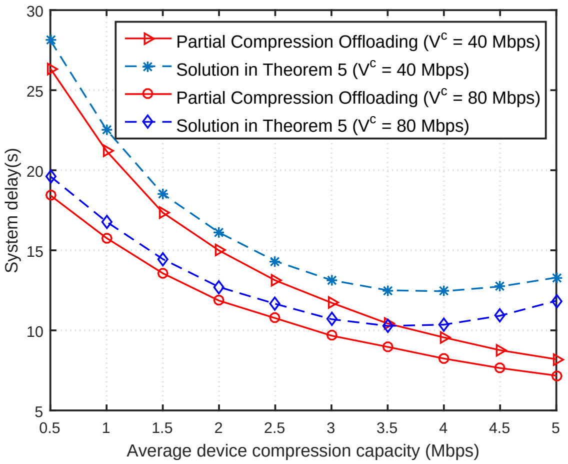
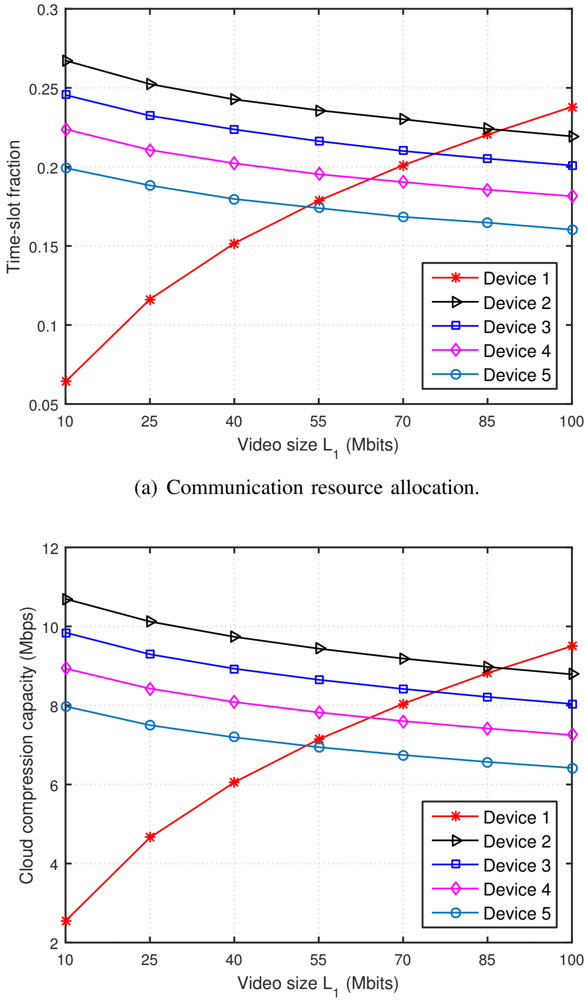


Fig. 4. Comparison between the partial compression offloading and the solution in Theorem 5.

Fig. 3(b) shows the minimum system delay versus the average device compression capacity in the three different models. In this simulation, we assume 20 mobile devices in the system while varying the average local compression capacity of all devices from 0.75 Mbps to 2 Mbps. From the figure, we can observe that the system delays of the local compression and partial compression offloading models decrease with the average device compression capacity since both use the local computation resource for video compression. Furthermore, the solution in Theorem 5 has a very close-to-optimal performance especially when the device compression capacity is small, demonstrating its accuracy and applicability in our system.

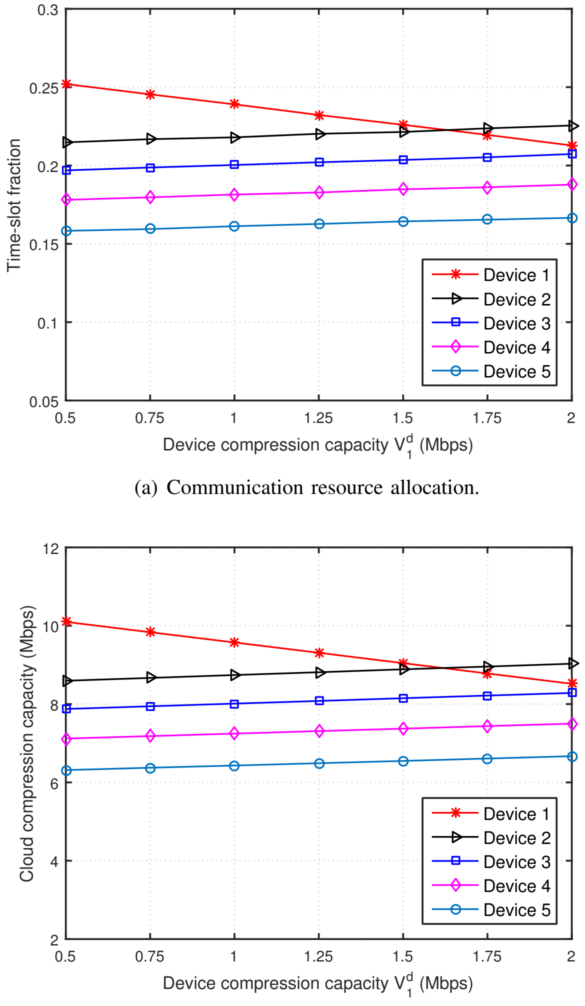
Fig. 4 illustrates the minimum system delay versus the average device compression capacity in both the partial compression offloading model and the solution in Theorem 5, where the communication resource B is 10 MHz. From this figure, we can observe that the delay gap between the optimal partial compression offloading and the solution in Theorem 5 becomes more evident as the average device compression capacity grows, which consists with the prerequisite that Theorem 5 can only be applied in the communication dominant system. Specifically, the system delay of the solution in Theorem 5 first decreases and then increases with the device compression capacity. The reason is that, when the device compression capacity becomes large, the resource allocation given in Theorem 5 no longer achieves a near-optimal performance and thus the video segmentation strategy in Lemma 2 is not suitable for such cases.



(a) Communication resource allocation.

(b) Computation resource allocation.

Fig. 5. Optimal resource allocation with different video sizes.



(a) Communication resource allocation.

(b) Computation resource allocation.

Fig. 6. Optimal resource allocation with different local compression capacities.

B. Optimal Resource Allocation in Partial Compression Offloading Model

Next, we analyze the impact of video size and device compression capacity on the optimal resource allocation in the partial compression offloading model. In this simulation, we assume five devices in the system and keep the video size and local compression capacity of devices 2-5 fixed while varying those parameters of device 1. The detailed simulation parameters for all devices are summarized in Table III.

TABLE III
SIMULATION PARAMETERS OF FIVE DEVICES

Device	Video Size	Local compression capacity
1	10-100 Mbits	0.5-2 Mbps
2	90 Mbits	1.2 Mbps
3	80 Mbits	1.3 Mbps
4	70 Mbits	1.4 Mbps
5	60 Mbits	1.5 Mbps

Fig. 5 illustrates the optimal time-slot and cloud computation resource allocations with different video sizes of device 1, where the local compression capacity of device 1 is fixed to 1.1 Mbps. It can be observed that the optimal resources t_1^* and V_1^{c*} allocated to device 1 increase with its video size. On the other aspect, the resources assigned to other devices will consequently decrease. This is rather intuitive due to the fact that more resources should be allocated to device 1 to minimize the system delay as its video size increases. Furthermore, it is shown that the optimal communication and computation resource allocations have almost the same trend, as displayed in Fig. 5(a) and Fig. 5(b). The reason for this outcome is clear since the communication and computation resources have the same effect on computing the end-to-end delay of each device.

Fig. 6 presents the optimal time-slot and edge cloud compression capacity with different local compression capacities of device 1, where the video size of device 1 is fixed to 100 Mbits. It can be observed that the optimal resources t_1^* and V_1^{c*} allocated to device 1 decrease with its local compression capacity. On the other aspect, the resources allocated to other devices will consequently increase. The reason is that, more resources should be allocated to those devices with lower compression capacity to reduce the weighted-sum delay of all devices. In addition, both optimal communication and computation resource allocations have an approximately linear trend, as shown in Fig. 6(a) and Fig. 6(b). This is due to the fact that the communication capacity $t_k R_k$, the device compression capacity V_k^{d}, and the cloud compression capacity V_k^{c} have the same effect on calculating the end-to-end delay of each device.

VII. Conclusion

This paper investigates joint communication and computation resource allocation for a TDMA-based multi-user MECO system. Our optimization aims to improve the QoE for users by minimizing the weighted-sum delay of all devices. Three models, namely local compression, edge cloud compression, and partial compression offloading, are studied and compared. The optimal solutions for both local compression and edge cloud compression are firstly achieved in closed-form, and some inherent insights are also highlighted. In the partial compression offloading model, we derive the optimal video segmentation strategy for each device in a closed-form expression. Then we formulate a piecewise convex optimization problem, which can be efficiently solved by a developed subgradient method. Moreover, to gain some insights, we consider a specific scenario in which communication capacity is much greater than device compression capacity. In this specific scenario, the closed-form solution can be derived. Finally, numerical results demonstrate that the partial compression offloading can efficiently reduce the end-to-end latency as compared against the other two models.

Future works may consider latency-minimization communication and computation resource allocation problem with non-orthogonal channel access where co-channel interference exists. In such a case, our analytical result for the delay performance remains unchanged but the channel capacity expression for each device will be more complicated. Non-convex optimization tools should be utilized to deal with this scenario. Another interesting direction of our future work is to investigate energy-efficiency optimization problem for the multiuser MECO system, i.e., minimizing the overall energy consumption by jointly allocating communication and computation resources. Moreover, some interesting issues, such as imperfect CSI and dynamical resource allocation also deserve further investigation in the future work.

Appendix A
Proof of Lemma 1

To prove Lemma 1, we first calculate a critical case for device k that satisfies

$$\begin{cases} D_{\mathrm{comp},k}^{\mathrm{d}} = D_{\mathrm{tran},k}^{\mathrm{c}}, \\ D_{\mathrm{tran},k}^{\mathrm{d}} = D_{\mathrm{comp},k}^{\mathrm{c}}, \end{cases} \tag{28}$$

which means that the delay for compressing the local part data equals to that for transmitting the edge cloud part data, and the delay for compressing the edge cloud part data also equals to that for transmitting the locally compressed data. Then applying the detailed delay expressions into (28), we can obtain the condition to reach this critical case, as

$$t_k R_k = \sqrt{\beta V_k^{\mathrm{d}} V_k^{\mathrm{c}}}. \tag{29}$$

In addition, let us define $\lambda_k^{(1)} = \frac{V_k^{\mathrm{d}}}{t_k R_k + V_k^{\mathrm{d}}}$ which satisfies $D_{\mathrm{comp},k}^{\mathrm{d}} = D_{\mathrm{tran},k}^{\mathrm{c}}$ and $\lambda_k^{(2)} = \frac{t_k R_k}{t_k R_k + \beta V_k^{\mathrm{c}}}$ which satisfies $D_{\mathrm{tran},k}^{\mathrm{d}} = D_{\mathrm{comp},k}^{\mathrm{c}}$. Then we can prove Lemma 1 by the following analysis.

A. Case A: $t_k R_k \geq \sqrt{\beta V_k^d V_k^c}$

In this case, $\lambda_k^{(1)} \leq \lambda_k^{(2)}$. When $\lambda_k \in \left[0, \lambda_k^{(1)}\right)$, $D_{\mathrm{comp},k}^{\mathrm{d}} < D_{\mathrm{tran},k}^{\mathrm{c}}$ and $D_{\mathrm{tran},k}^{\mathrm{d}} < D_{\mathrm{comp},k}^{\mathrm{c}}$. Therefore, we have $D_k = D_{\mathrm{tran},k}^{\mathrm{c}} + D_{\mathrm{comp},k}^{\mathrm{c}} = \frac{(1-\lambda_k)L_k}{t_k R_k} + \frac{(1-\lambda_k)L_k}{V_k^{\mathrm{c}}}$, which decreases with λ_k. When $\lambda_k \in \left[\lambda_k^{(1)}, \lambda_k^{(2)}\right]$, $D_{\mathrm{comp},k}^{\mathrm{d}} \geq D_{\mathrm{tran},k}^{\mathrm{c}}$ and $D_{\mathrm{tran},k}^{\mathrm{d}} \leq D_{\mathrm{comp},k}^{\mathrm{c}}$. Thus we have $D_k = \max(D_{\mathrm{comp},k}^{\mathrm{d}} + D_{\mathrm{tran},k}^{\mathrm{d}}, D_{\mathrm{tran},k}^{\mathrm{c}} + D_{\mathrm{comp},k}^{\mathrm{c}})$. Since $D_{\mathrm{comp},k}^{\mathrm{d}} + D_{\mathrm{tran},k}^{\mathrm{d}} = \frac{\lambda_k L_k}{V_k^{\mathrm{d}}} + \frac{\beta \lambda_k L_k}{t_k R_k}$ increases with λ_k while $D_{\mathrm{tran}}^{\mathrm{c}} + D_{\mathrm{comp},k}^{\mathrm{c}} = \frac{(1-\lambda_k)L_k}{t_k R_k} + \frac{(1-\lambda_k)L_k}{V_k^{\mathrm{c}}}$ decreases with λ_k, the delay of device k achieves to the minimum when $D_{\mathrm{comp},k}^{\mathrm{d}} + D_{\mathrm{tran},k}^{\mathrm{d}} = D_{\mathrm{tran},k}^{\mathrm{c}} + D_{\mathrm{comp},k}^{\mathrm{c}}$, which results in $\lambda_k = \frac{V_k^{\mathrm{d}}(t_k R_k + V_k^{\mathrm{c}})}{V_k^{\mathrm{d}} V_k^{\mathrm{c}}(1+\beta) + t_k R_k (V_k^{\mathrm{d}} + V_k^{\mathrm{c}})} \in \left[\lambda_k^{(1)}, \lambda_k^{(2)}\right]$. Finally when $\lambda_k \in \left(\lambda_k^{(2)}, 1\right]$, $D_{\mathrm{comp},k}^{\mathrm{d}} > D_{\mathrm{tran},k}^{\mathrm{c}}$ and $D_{\mathrm{tran},k}^{\mathrm{d}} > D_{\mathrm{comp},k}^{\mathrm{c}}$. Therefore, we have $D_k = D_{\mathrm{comp},k}^{\mathrm{d}} + D_{\mathrm{tran},k}^{\mathrm{d}} = \frac{\lambda_k L_k}{V_k^{\mathrm{d}}} + \frac{\beta \lambda_k L_k}{t_k R_k}$, which increases with λ_k. Based on the above analysis, the optimal video segmentation strategy in this case is $\lambda_k^* = \frac{V_k^{\mathrm{d}}(t_k R_k + V_k^{\mathrm{c}})}{V_k^{\mathrm{d}} V_k^{\mathrm{c}}(1+\beta) + t_k R_k (V_k^{\mathrm{d}} + V_k^{\mathrm{c}})}$.

B. Case B: $t_k R_k < \sqrt{\beta V_k^d V_k^c}$

In this case, $\lambda_k^{(1)} > \lambda_k^{(2)}$. When $\lambda_k \in \left[0, \lambda_k^{(2)}\right)$, we have $D_{\mathrm{comp},k}^{\mathrm{d}} < D_{\mathrm{tran},k}^{\mathrm{c}}$ and $D_{\mathrm{tran},k}^{\mathrm{d}} < D_{\mathrm{comp},k}^{\mathrm{c}}$. Therefore, we have $D_k = D_{\mathrm{tran},k}^{\mathrm{c}} + D_{\mathrm{comp},k}^{\mathrm{c}} = \frac{(1-\lambda_k)L_k}{t_k R_k} + \frac{(1-\lambda_k)L_k}{V_k^{\mathrm{c}}}$, which decreases with λ_k. When $\lambda_k \in \left[\lambda_k^{(2)}, \lambda_k^{(1)}\right]$, $D_{\mathrm{comp},k}^{\mathrm{d}} \leq D_{\mathrm{tran},k}^{\mathrm{c}}$ and $D_{\mathrm{tran},k}^{\mathrm{d}} \geq D_{\mathrm{comp},k}^{\mathrm{c}}$. Thus we have $D_k = D_{\mathrm{tran},k}^{\mathrm{c}} + D_{\mathrm{tran},k}^{\mathrm{d}} = \frac{L_k}{t_k R_k}\left(1 + (\beta - 1)\lambda_k\right)$, which also decreases with λ_k because $0 < \beta < 1$. Finally when $\lambda_k \in \left(\lambda_k^{(1)}, 1\right]$, $D_{\mathrm{comp},k}^{\mathrm{d}} > D_{\mathrm{tran},k}^{\mathrm{c}}$ and $D_{\mathrm{tran},k}^{\mathrm{d}} > D_{\mathrm{comp},k}^{\mathrm{c}}$. Therefore, we have $D_k = D_{\mathrm{comp},k}^{\mathrm{d}} + D_{\mathrm{tran},k}^{\mathrm{d}} = \frac{\lambda_k L_k}{V_k^{\mathrm{d}}} + \frac{\beta \lambda_k L_k}{t_k R_k}$, which increases with λ_k. Based on the above discussion, the optimal video segmentation strategy in this case is $\lambda_k^* = \lambda_k^{(1)} = \frac{V_k^{\mathrm{d}}}{V_k^{\mathrm{d}} + t_k R_k}$.

Appendix B
Proof of Theorem 3

Note that all constraints in Problem 4 are affine. Therefore, Problem 4 is convex if the objective function is convex. In the following, we first prove that $\widehat{D}_k$ is a continuously piecewise convex function.

$$\begin{aligned}\Delta_1 = H_{11} = 2L_k \Bigg(& \frac{\beta(1+\beta)\left(V_k^{\mathrm{d}}V_k^{\mathrm{c}}\right)^2\left((1+\beta)V_k^{\mathrm{d}}V_k^{\mathrm{c}}+3t_kR_k\left(V_k^{\mathrm{d}}+V_k^{\mathrm{c}}\right)\right)}{t_k^3R_k\left((1+\beta)\,V_k^{\mathrm{d}}V_k^{\mathrm{c}}+t_kR_k\left(V_k^{\mathrm{d}}+V_k^{\mathrm{c}}\right)\right)^3}+ \\ & +\frac{(t_kR_k)^2\left(V_k^{\mathrm{d}}+V_k^{\mathrm{c}}\right)\left(3\beta V_k^{\mathrm{d}}V_k^{\mathrm{c}}\left(V_k^{\mathrm{d}}+V_k^{\mathrm{c}}\right)+t_kR_k\left(\beta\left(V_k^{\mathrm{d}}\right)^2+\left(V_k^{\mathrm{c}}\right)^2\right)\right)}{t_k^3R_k\left((1+\beta)\,V_k^{\mathrm{d}}V_k^{\mathrm{c}}+t_kR_k\left(V_k^{\mathrm{d}}+V_k^{\mathrm{c}}\right)\right)^3}\Bigg)>0.\end{aligned} \tag{30}$$

$$\begin{aligned}\Delta_2 =& H_{11}H_{22}-H_{12}H_{21} \\ =& 4L_k^2\left(t_kR_k+\beta V_k^{d}\right)^2\Bigg(\frac{\beta(1+\beta)\left(V_k^{\mathrm{d}}\right)^2V_k^{\mathrm{c}}\left((1+\beta)V_k^{\mathrm{d}}V_k^{\mathrm{c}}+t_kR_k\left(2V_k^{\mathrm{d}}+3V_k^{\mathrm{c}}\right)\right)}{t_k^3R_k\left((1+\beta)V_k^{\mathrm{d}}V_k^{\mathrm{c}}+t_kR_k\left(V_k^{\mathrm{d}}+V_k^{\mathrm{c}}\right)\right)^5} \\ & +\frac{(t_kR_k)^2\left(t_kR_k\left(\beta\left(V_k^{\mathrm{d}}\right)^2+\left(V_k^{\mathrm{c}}\right)^2\right)+\beta V_k^{\mathrm{d}}\left(\left(V_k^{\mathrm{d}}\right)^2+4V_k^{\mathrm{d}}V_k^{\mathrm{c}}+3\left(V_k^{\mathrm{c}}\right)^2\right)\right)}{t_k^3R_k\left((1+\beta)V_k^{\mathrm{d}}V_k^{\mathrm{c}}+t_kR_k\left(V_k^{\mathrm{d}}+V_k^{\mathrm{c}}\right)\right)^5}\Bigg)>0.\end{aligned} \tag{31}$$

The Hessian of $\widehat{D}_{k,1}$ is

$$\mathbf{H}=\left[\begin{array}{cc}H_{11} & H_{12}\\ H_{21} & H_{22}\end{array}\right]=\left[\begin{array}{cc}\dfrac{\partial^2\widehat{D}_{k,1}}{\partial t_k^2} & \dfrac{\partial^2\widehat{D}_{k,1}}{\partial t_k\partial V_k^{\mathrm{c}}}\\ \dfrac{\partial^2\widehat{D}_{k,1}}{\partial V_k^{\mathrm{c}}\partial t_k} & \dfrac{\partial^2\widehat{D}_{k,1}}{\partial (V_k^{\mathrm{c}})^2}\end{array}\right]. \tag{32}$$

We can prove that the Hessian in (32) is positive-definite by proving all the leading principal minors of $\mathbf{H}$ are positive, which are given by (30) and (31), as shown at the top of this page. Therefore, $\widehat{D}_{k,1}$ is strictly convex on both t_k and V_k^{c}.

Next, we prove that $\widehat{D}_{k,2}$ is also convex on t_k and V_k^{c}. The second-order partial derivative of $\widehat{D}_{k,2}$ on t_k fulfills

$$\begin{aligned}&\frac{\partial^2\widehat{D}_{k,2}}{\partial t_k^2}= \\ &\frac{2L_k\left((t_kR_k)^3+3\beta V_k^{\mathrm{d}}(t_kR_k)^2+3\beta\left(V_k^{\mathrm{d}}\right)^2t_kR_k+\beta\left(V_k^{\mathrm{d}}\right)^3\right)}{t_k^3R_k\left(t_kR_k+V_k^{\mathrm{d}}\right)^3}>0.\end{aligned} \tag{33}$$

Since $\widehat{D}_{k,2}$ does not change over V_k^{c}, $\widehat{D}_{k,2}$ is convex on t_k and V_k^{c}. Moreover, it can be easily verified that $\widehat{D}_{k,1}=\widehat{D}_{k,2}$, $\dfrac{\partial\widehat{D}_{k,1}}{\partial t_k}\neq\dfrac{\partial\widehat{D}_{k,2}}{\partial t_k}$, and $\dfrac{\partial\widehat{D}_{k,1}}{\partial V_k^{\mathrm{c}}}\neq\dfrac{\partial\widehat{D}_{k,2}}{\partial V_k^{\mathrm{c}}}$ at $t_kR_k=\sqrt{\beta V_k^{\mathrm{d}}V_k^{\mathrm{c}}}$. Therefore $\widehat{D}_k$ is a continuous and piecewise function. Then computing the difference between $\widehat{D}_{k,1}$ and $\widehat{D}_{k,2}$, we have

$$\begin{aligned}&\widehat{D}_{k,1}-\widehat{D}_{k,2}= \\ &\frac{L_k}{t_kR_k}\frac{\left(t_kR_k+\beta V_k^{\mathrm{d}}\right)\left((t_kR_k)^2-\beta V_k^{\mathrm{d}}V_k^{\mathrm{c}}\right)}{\left(V_k^{\mathrm{d}}V_k^{\mathrm{c}}(1+\beta)+t_kR_k\left(V_k^{\mathrm{d}}+V_k^{\mathrm{c}}\right)\right)\left(t_kR_k+V_k^{\mathrm{d}}\right)}.\end{aligned} \tag{34}$$

Therefore, when $t_kR_k\geq\sqrt{\beta V_k^{\mathrm{d}}V_k^{\mathrm{c}}}$, $\widehat{D}_{k,1}\geq\widehat{D}_{k,2}$, otherwise $\widehat{D}_{k,1}<\widehat{D}_{k,2}$. According to (16), $\widehat{D}_k$ can be rewritten as $\max\{\widehat{D}_{k,1},\widehat{D}_{k,2}\}$, which is convex on t_k and V_k^{c} since the pointwise maximum preserves convexity. Furthermore, the objective function $\sum_{k=1}^K\alpha_k\widehat{D}_k$ is the summation of a set of convex functions, which is also convex. This ends the proof.

Appendix C
Proof of Theorem 4

In the following, we will prove that the iterative method in Theorem 4 could converge to the optimal resource allocation solution, denoted as $\boldsymbol{x}^*$. First, the Euclidean distance between the $(n+1)^{\mathrm{th}}$ iteration solution and the optimal solution can be calculated as

$$\left|\left|\boldsymbol{x}^{(n+1)}-\boldsymbol{x}^*\right|\right|_2^2=\left|\left|\boldsymbol{x}^{(n)}-\phi_n\boldsymbol{g}^{(n)}-\boldsymbol{x}^*\right|\right|_2^2 \tag{35}$$

$$=\left|\left|\boldsymbol{x}^{(n)}-\boldsymbol{x}^*\right|\right|_2^2-2\phi_n\boldsymbol{g}^{(n)T}\left(\boldsymbol{x}^{(n)}-\boldsymbol{x}^*\right)+\phi_n^2\left|\left|\boldsymbol{g}^{(n)}\right|\right|_2^2 \tag{36}$$

$$\leq\left|\left|\boldsymbol{x}^{(n)}-\boldsymbol{x}^*\right|\right|_2^2-2\phi_n\left(F(\boldsymbol{x}^{(n)})-F\left(\boldsymbol{x}^*\right)\right)+\phi_n^2\left|\left|\boldsymbol{g}^{(n)}\right|\right|_2^2 \tag{37}$$

$$\leq\left|\left|\boldsymbol{x}^{(0)}-\boldsymbol{x}^*\right|\right|_2^2-2\sum_{i=0}^n\phi_i\left(F(\boldsymbol{x}^{(i)})-F(\boldsymbol{x}^*)\right)+\sum_{i=0}^n\phi_i^2\left|\left|\boldsymbol{g}^{(i)}\right|\right|_2^2, \tag{38}$$

where $||\boldsymbol{x}||_2$ is the Euclidean norm of $\boldsymbol{x}$ and $\boldsymbol{g}^{(n)T}$ represents the transpose of $\boldsymbol{g}^{(n)}$. Notice that the inequality operation (37) is based on the convexity of $\sum_{k=1}^K\alpha_k\widehat{D}_k$, as $F(\boldsymbol{x}^*)\geq F(\boldsymbol{x}^{(n)})+\boldsymbol{g}^{(n)T}\left(\boldsymbol{x}^*-\boldsymbol{x}^{(n)}\right)$.

Let us denote $F_{\text{best}}^{(n)}=\min_{i=0}^nF\left(\boldsymbol{x}^{(i)}\right)$. Then we have

$$\sum_{i=0}^n\phi_i\left(F\left(\boldsymbol{x}^{(i)}\right)-F\left(\boldsymbol{x}^*\right)\right)\geq\left(F_{\text{best}}^{(n)}-F\left(\boldsymbol{x}^*\right)\right)\left(\sum_{i=0}^n\phi_i\right). \tag{39}$$

Substituting (39) into (38), we can derive the upper bound difference between $F_{\text{best}}^{(n)}$ and $F(\boldsymbol{x}^*)$ as follows.

$$F_{\text{best}}^{(n)}-F\left(\boldsymbol{x}^*\right) \tag{40}$$

$$\leq\frac{\left|\left|\boldsymbol{x}^{(0)}-\boldsymbol{x}^*\right|\right|_2^2-\left|\left|\boldsymbol{x}^{(n+1)}-\boldsymbol{x}^*\right|\right|_2^2+\sum_{i=0}^n\phi_i^2\left|\left|\boldsymbol{g}^{(i)}\right|\right|_2^2}{2\sum_{i=0}^n\phi_i} \tag{41}$$

$$\leq\frac{\left|\left|\boldsymbol{x}^{(0)}-\boldsymbol{x}^*\right|\right|_2^2-\left|\left|\boldsymbol{x}^{(n+1)}-\boldsymbol{x}^*\right|\right|_2^2+G\sum_{i=0}^n\phi_i^2}{2\sum_{i=0}^n\phi_i} \tag{42}$$

$$\leq\frac{\left|\left|\boldsymbol{x}^{(0)}-\boldsymbol{x}^*\right|\right|_2^2+G\sum_{i=0}^n\phi_i^2}{2\sum_{i=0}^n\phi_i} \tag{43}$$

$$\leq\frac{R+G\sum_{i=0}^n\phi_i^2}{2\sum_{i=0}^n\phi_i}\leq\epsilon, \tag{44}$$

where $R=\sup\limits_{\boldsymbol{x}}\left|\left|\boldsymbol{x}^{(0)}-\boldsymbol{x}\right|\right|_2^2$, $G=\max_{i=0}^n\left|\left|\boldsymbol{g}^{(i)}\right|\right|_2^2$, and ϵ denotes the maximum convergence tolerance.

Under such circumstances, when we select $\phi_i=\sqrt{\dfrac{R}{G\cdot n}}$,

$(i = 0, 1, \cdots, n)$, the required number of iterations $n = \dfrac{RG}{\epsilon^2}$. Moreover, $F_{\text{best}}^{(n)} - F(\boldsymbol{x}^*)$ will gradually converge to zero and the iteration will linearly converge to the optimal solution $\boldsymbol{x}^*$ when ϕ_i satisfies $\sum_{i=0}^{\infty} \phi_i = \infty$ and $\sum_{i=0}^{\infty} \phi_i^2 < \infty$. This can be realized by selecting the Polyak step size $\phi_i = \dfrac{F\left(\boldsymbol{x}^{(i)}\right) - F_{\text{best}}^{(i)} + \gamma_i}{\left|\left|\boldsymbol{g}^{(i)}\right|\right|_2^2}$, where γ_i satisfies $\sum_{i=0}^{\infty} \gamma_i = \infty$ and $\sum_{i=0}^{\infty} \gamma_i^2 < \infty$ [25]. This ends the proof.

Appendix D
Proof of Theorem 5

It can be easily verified that the end-to-end delay expression (24) is strictly convex on t_k and V_k^{c} using the derivation method in Appendix B. Therefore, we can utilize the KKT conditions to derive the closed-form solution. Let $\{\bar{t}_k^*, \overline{V}_k^{\text{c}*}\}$ denote the optimal solution for this specific scenario. Then the Lagrange function of $\sum_{k=1}^{K} \alpha_k \overline{D}_k$ can be expressed as

$$\overline{L}_{\text{P}} = \sum_{k=1}^{K} \alpha_k \overline{D}_k + \theta\left(\sum_{k=1}^{K} t_k - 1\right) + \omega\left(\sum_{k=1}^{K} V_k^{\text{c}} - V^{\text{c}}\right). \quad (45)$$

Applying the KKT conditions leads to the following necessary and sufficient conditions

$$\frac{\partial \overline{L}_{\text{P}}}{\partial \bar{t}_k^*} = -\frac{\alpha_k L_k R_k \left(\overline{V}_k^{\text{c}*}\right)^2}{\left(\bar{t}_k^* R_k V_k^{\text{d}} + \bar{t}_k^* R_k \overline{V}_k^{\text{c}*} + V_k^{\text{d}} \overline{V}_k^{\text{c}*}\right)^2} + \theta^* \begin{cases} \geq 0, & \bar{t}_k^* = 0, \\ = 0, & \bar{t}_k^* > 0, \end{cases} \quad (46)$$

$$\frac{\partial \overline{L}_{\text{P}}}{\partial \overline{V}_k^{\text{c}*}} = -\frac{\alpha_k L_k \left(\bar{t}_k^* R_k\right)^2}{\left(\bar{t}_k^* R_k V_k^{\text{d}} + \bar{t}_k^* R_k \overline{V}_k^{\text{c}*} + V_k^{\text{d}} \overline{V}_k^{\text{c}*}\right)^2} + \omega^* \begin{cases} \geq 0, & \overline{V}_k^{\text{c}*} = 0, \\ = 0, & \overline{V}_k^{\text{c}*} > 0, \end{cases} \quad (47)$$

$$\theta^*\left(\sum_{k=1}^{K} \bar{t}_k^* - 1\right) = 0, \ \sum_{k=1}^{K} \bar{t}_k^* \leq 1, \ \theta^* \geq 0, \quad (48)$$

$$\omega^*\left(\sum_{k=1}^{K} \overline{V}_k^{\text{c}*} - V^{\text{c}}\right) = 0, \ \sum_{k=1}^{K} \overline{V}_k^{\text{c}*} \leq V^{\text{c}}, \ \omega^* \geq 0. \quad (49)$$

Based on the above conditions, we can derive the optimal resource allocation solution for the special case, as

$$\begin{cases} \bar{t}_k^* = \dfrac{\overline{V}_k^{\text{c}*}\left(\sqrt{\dfrac{\alpha_k L_k R_k}{\theta^*}} - V_k^{\text{d}}\right)^+}{R_k\left(V_k^{\text{d}} + \overline{V}_k^{\text{c}*}\right)}, \ \forall k \in \mathcal{K}, \\ \overline{V}_k^{\text{c}*} = \dfrac{\bar{t}_k^* R_k\left(\sqrt{\dfrac{\alpha_k L_k}{\omega^*}} - V_k^{\text{d}}\right)^+}{\bar{t}_k^* R_k + V_k^{\text{d}}}, \ \forall k \in \mathcal{K}. \end{cases} \quad (50)$$

This ends the proof.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswmi, "Internet of Things (IoT): A vision, architectural elements, and future directions," *ELSEVIER Future Gener. Compt. Syst.*, vol. 29, no. 7, pp. 1645-1660, Sep. 2013.

[2] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854-864, Dec. 2016.

[3] CISCO, "The Internet of Things how the next evolution of the Internet is changing everything," White paper, Apr. 2011. [Online]. Available: http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.

[4] European Telecommunications Standards Institute, "Mobile-edge-computing-Introductory technical white paper," Sep. 2014. [Online]. Available: https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_introductory_technical_white_paper_v1%2018-09-14.pdf.

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surv. Tut.*, vol. 19, no. 4, pp. 2322-2358, Aug. 2017.

[6] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590-3605, Dec. 2016.

[7] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 1451-1455.

[8] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569-4581, Sep. 2013.

[9] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE J. Select. Areas Commun.*, vol. 34, no. 5, pp. 1757-1771, May 2016.

[10] Y. H. Kao, B. Krishnamachari, M. R. Ra, and F. Bai, "Hermes: Latency optimal task assignment for resource-constrained mobile computing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Hong Kong, China, Apr. 2015, pp. 1894-1902.

[11] S. E. Mahmoodi, R. N. Uma, and K. P. Subbalakshmi, "Optimal joint scheduling and cloud offloading for mobile applications," *IEEE Trans. Cloud Comput.*, vol. PP, no. 99, pp. 1-13, 2016.

[12] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268-4282, Oct. 2016.

[13] M. Jia, J. Cao, and L. Yang, "Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM WKSHPS)*, Toronto, Canada, Apr. 2014, pp. 352-357.

[14] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397-1411, Mar. 2017.

[15] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994-6009, Sep. 2017.

[16] 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. (GLOBECOM)*, Washington, DC, Dec. 2016, pp. 1-6.

[17] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE Trans. Netw.*, vol. 24, no. 5, pp. 2795-2808, Oct. 2016.

[18] M. Molina, O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Joint scheduling of communication and computation resources in multiuser wireless application offloading," in *Proc. IEEE Int. Symp. on Personal Indoor and Mobile Radio Comm. (PIMRC)*, Washington, DC, Sep. 2014, pp. 1093-1098.

[19] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89-103, Jun. 2015.

[20] Y. H. Kao, B. Krishnamachari, M. R. Ra, and F. Bai, "Hermes: Latency optimal task assignment for resource-constrained mobile computing," *IEEE Trans Mobile Comput.*, vol. 16, no. 11, pp. 3056-3069, Nov. 2017.

[21] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784-1797, Mar. 2018.

[22] Z. Chang et al., "Energy efficient resource allocation for wireless power transfer enabled collaborative mobile clouds," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3438-3450, Dec. 2016.

[23] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Content caching at the wireless network edge: A distributed algorithm via belief propagation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1-6.

[24] David Williams, *Probability with Martingale*. Cambridge University Press, 1991.

[25] S. Boyd and J. Duchi, “Convex Optimization II” Standford Univ., USA, Lect. Notes EE364b, Spring Quart. 2008.

[26] L. Vandenberghe, “Optimization Methods for Large-Scale Systems” UCLA Univ., USA, Lect. Notes EE236C, Spring Quart. 2016.

[27] H. Kruegle, *CCTV Surveillance: Analog and Digital Video Practices and Technology,* Second Edition, Elsevier, 2007, pp. 296-298.

[28] L. Hanzo, P. Cherriman, J. Streit, *Video Compression and Communications*, Wiley-IEEE Press, 2008, pp. 379-406.

Jinke Ren received the B.S.E degree in information engineering from Zhejiang University, Hangzhou, China, in 2017. He is currently working toward the Ph.D degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. His current research interests mainly include machine learning and mobile edge computing.

Guanding Yu (S’05-M’07-SM’13) received the B.E. and Ph.D. degrees in communication engineering from Zhejiang University, Hangzhou, China, in 2001 and 2006, respectively. He joined Zhejiang University in 2006, and is now a Full Professor with the College of Information and Electronic Engineering. From 2013 to 2015, he was also a Visiting Professor at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include 5G communications and networks, mobile edge computing, and machine learning for wireless networks.

Dr. Yu has served as a guest editor of *IEEE Communications Magazine* special issue on Full-Duplex Communications, an editor of *IEEE Journal on Selected Areas in Communications* Series on Green Communications and Networking, an associate editor of *IEEE Access*, and a lead guest editor of *IEEE Wireless Communications Magazine* special issue on LTE in Unlicensed Spectrum. He is now serving as an editor of *IEEE Transactions on Green Communications and Networking* and an editor of *IEEE Wireless Communications Letters*. He received the 2016 Exemplary Reviewer Award from the *IEEE Transactions on Communications* of the IEEE Communications Society (ComSoc) and the 2016 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award. He regularly sits on the technical program committee (TPC) boards of prominent IEEE conferences such as ICC, GLOBECOM, and VTC. He has also served as the General Co-Chair for the CRNet 2010 Workshop, a TPC Co-Chair for the Green Communications and Computing Symposium of Chinacom 2013.

Yunlong Cai (S’07-M’10-SM’16) received the B.S. degree in computer science from Beijing Jiaotong University, Beijing, China, in 2004, the M.Sc. degree in electronic engineering from the University of Surrey, Guildford, U.K., in 2006, and the Ph.D. degree in electronic engineering from the University of York, York, U.K., in 2010. From February 2010 to January 2011, he was a Postdoctoral Fellow at the Electronics and Communications Laboratory of the Conservatoire National des Arts et Metiers (CNAM), Paris, France. Since February 2011, he has been with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, where he is currently an Associate Professor. From August 2016 to January 2017, he was a Visiting Scholar at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include transceiver design for multiple-antenna systems, sensor array processing, mmWave communications, full-duplex communications, and cooperative and relay communications.

Yinghui He received the B.S.E degree in information engineering from Zhejiang University, Hangzhou, China, in 2018. He is currently working toward the master degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. His research interests mainly include mobile edge computing and device-to-device communications.