Collaborative Cloud and Edge Computing for Latency Minimization

Jinke Ren, Guanding Yu, *Senior Member, IEEE*, Yinghui He, and Geoffrey Ye Li, *Fellow, IEEE*

***Abstract*—By performing data processing at the network edge, mobile edge computing can effectively overcome the deficiencies of network congestion and long latency in cloud computing systems. To improve edge cloud efficiency with limited communication and computation capacities, we investigate the collaboration between cloud computing and edge computing, where the tasks of mobile devices can be partially processed at the edge node and at the cloud server. First, a joint communication and computation resource allocation problem is formulated to minimize the weighted-sum latency of all mobile devices. Then, the closed-form optimal task splitting strategy is derived as a function of the normalized backhaul communication capacity and the normalized cloud computation capacity. Some interesting and useful insights for the optimal task splitting strategy are also highlighted by analyzing four special scenarios. Based on this, we further transform the original joint communication and computation resource allocation problem into an equivalent convex optimization problem and obtain the closed-form computation resource allocation strategy by leveraging the convex optimization theory. Moreover, a necessary condition is also developed to judge whether a task should be processed at the corresponding edge node only, without offloading to the cloud server. Finally, simulation results confirm our theoretical analysis and demonstrate that the proposed collaborative cloud and edge computing scheme can evidently achieve a better delay performance than the conventional schemes.**

***Index Terms*—Mobile edge computing, mobile cloud computing, latency minimization, joint resource allocation, task splitting strategy, collaborative cloud and edge computing.**

I. Introduction

To cope with the tremendous growth of mobile data traffic, edge computing has been recently proposed as a promising means to substantially improve the computation performance by deploying cloud computing services at the edge of a network [2]–[4]. This technique can effectively overcome the deficiencies of core network congestion and long transmission latency in traditional cloud computing systems. In the context of mobile cellular networks, the edge computing services can be implemented at the base stations (BSs), which is also known as mobile edge computing (MEC) [5], [6].

Manuscript received August 20, 2018; revised February 21, 2019; accepted March 07, 2019. This work was supported in part by the Natural Science Foundation of China under Grant 61671407, the Open Research Fund of the State Key Laboratory of Integrated Services Networks, Xidian University, under Grant ISN18-13, and the Fundamental Research Funds for the Central Universities. This paper was presented in part at IEEE WCNC 2019, Marrakech, Morocco, April 2019 [1]. The associate editor coordinating the review of this paper and approving it for publication was Y. Guo. (*Corresponding author: G. Yu.*)

J. Ren, G. Yu, and Y. He 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, 2014hyh}@zju.edu.cn).

G. Y. Li is with the School of ECE, Georgia Institute of Technology, Atlanta, GA 0332-0250, USA (email: liye@ieee.org).

Although the edge computing has the great potential to relieve the burden on core networks, its main bottleneck is the limited computation and communication capacities as compared with the cloud computing. Therefore, in the cloud and edge coexistence system, hierarchical computing can be realized in which tasks can be opportunistically processed by both the edge node and the cloud server. For example, non-computational-intensive tasks can be processed at the edge node to achieve a lower end-to-end latency and better energy efficiency. On the other hand, it is better to offload computational-intensive tasks to the cloud server to take advantage of its abundant computation capacity. Therefore, effective collaboration between cloud computing and edge computing is of paramount importance to the performance improvement.

Meanwhile, the convergence of communication and computation is becoming an evident trend for future wireless networks. To this end, joint design of communication and computation in both cloud computing and edge computing has aroused much research interest in recent years. There are mainly two objectives for the joint resource allocation: maximizing the energy efficiency [7]–[13] and minimizing the end-to-end latency [14]–[23]. A theoretical framework for mobile cloud computing (MCC) systems under time-varying stochastic wireless channels was proposed in [7], where a threshold-based energy-efficient offloading policy between local computing and cloud computing was developed. This study was later extended in [8], where a closed-form optimal mode selection between local computing and cloud computing with the microwave power transfer technique was developed. A scalable approximate dynamic programming algorithm was developed in [9], which integrated the adaptive LTE/WiFi link selection and data transmission schedule to reduce the total mobile energy consumption in the MCC system. Moreover, to minimize the total energy consumption of mobile devices and cloud servers, the game-theoretical technique was adopted to design the distributed cloud server selection strategy [10]. The authors in [11] proposed a joint communication and computation resource allocation algorithm to minimize the device energy consumption while guaranteeing the offloading latency requirement in the multi-cell MEC system. Based on the game theory, a distributed computation offloading policy was devised in [12] for energy-delay tradeoff in the multi-user MEC system. Furthermore, the authors in [13] developed a threshold-based mobile offloading policy to maximize the energy efficiency of the multi-user MEC system.

The above works mainly focus on the energy efficiency improvement. However, end-to-end latency is another critical objective for the future wireless networks. In this respect, an

offline heuristic algorithm was developed in [14] to minimize the average latency for both program partitioning and task execution in the MCC system. The online task offloading method in [15] reduces the completion time of mobile applications by adopting a task-call graph to model the inter-dependency among different functions and routines in computation tasks. A polynomial-time approximate scheme and an online learning algorithm were proposed in [16] to minimize the end-to-end latency in a single-user MCC system under a prescribed resource utilization constraint. Based on the Markov decision process approach, the optimal computation task scheduling policy for both local computing and edge computing was developed in [17]. To improve the delay performance and reduce the failure probability of task execution, an optimal dynamic offloading policy based on the Lyapunov optimization was developed in [18]. A cooperative task offloading policy was proposed to deliver low latency performance for a multiuser MEC system by load sharing [19]. In [20], a joint computation offloading, content caching, and resource allocation strategy was developed to minimize the overall latency of an MEC system. Further, to meet the sensitive delay requirement for mobile applications, the joint computation partitioning and resource allocation problem was investigated in [21], and a heuristic algorithm was also developed therein. Moreover, to deal with the bursty task arrivals, the uplink and downlink transmission scheduling was integrated with MEC to reduce the average processing latency [22]. Last but not least, our previous work in [23] proposed a partial computation offloading scheme between the mobile devices and the edge cloud in a multi-user time-division multiple access (TDMA) MEC system, and developed an optimal closed-form data segmentation strategy to minimize the weighted-sum delay of all mobile devices.

As mentioned earlier, the system performance can be further improved if the cloud computing and edge computing can be effectively collaborated. Various fog-to-cloud hierarchical architectures were presented in [24]–[26] and joint task offloading and resource allocation algorithms for such collaborative computing systems were developed to achieve satisfactory latency and energy consumption performances [27]–[29]. However, the closed-form resource allocation results on how to collaborate the cloud and edge to achieve the optimal task processing performance have not been analyzed in the aforementioned works. Motivated by this, we consider the partial computation offloading model and joint communication and computation resource allocation for latency minimization in this paper. In particular, we investigate a hierarchical computing system within the multi-cell mobile cellular networks, where each BS is equipped with finite edge computing capacity. In such a system, the computation tasks of mobile devices can be processed at the corresponding edge server located at the BS or be transmitted to the cloud server deployed at the cloud center for processing. We will address the following two fundamental issues: 1) how to collaborate the edge node and the cloud server to achieve the optimal computing performance? 2) how to jointly allocate the communication and computation resources to minimize the end-to-end latency of mobile devices?

The main contributions of this work can be summarized as follows.

- To take full advantage of the edge computing capacity at BSs and the cloud computing capacity at the cloud server, we consider that the task of each mobile device can be partially processed at the edge node and partially offloaded to the cloud server for processing. With this regard, a joint communication and computation resource allocation problem is formulated to minimize the weighted-sum delay of all mobile devices. The optimal communication resource allocation is firstly derived in closed-form and some insightful results are also revealed.
- We find two significative parameters, i.e., the normalized backhaul communication capacity and the normalized cloud computation capacity, that affect the computation resource allocation of each mobile device. Therefore, we devise a closed-form optimal task splitting strategy as a function of these two parameters. To gain more insights, we further analyze four different network scenarios: communication-limited system, computation-limited system, edge-dominated system, and cloud-dominated system.
- Leveraging the optimal task splitting strategy, we convert the original joint communication and computation resource allocation problem into an equivalent convex optimization problem. Then, we obtain the closed-form computation resource allocation using the Karush-Kuhn-Tucker (KKT) conditions. Moreover, a necessary condition is further developed to judge whether a task should be processed at the corresponding edge node only, without offloading to the cloud server.

The rest of the paper is organized as follows. In Section II, we introduce the system model and formulate the latency-minimization problem. The optimal communication resource allocation, task splitting strategy, and computation resource allocation are investigated in Sections III, IV, and V, respectively. Numerical results are provided in Section VI and the paper is finally concluded in Section VII.

II. System Model and Problem Formulation

In this section, we will first describe the network architecture, communication model, and computation model of the cloud-edge collaboration system. After that, we will formulate the weighted-sum latency minimization problem.

A. System Model

As illustrated in Fig. 1, we consider a cloud-edge collaboration system with one centralized cloud server and J single-antenna BSs, denoted by a set $\mathcal{J} = \{1, 2, \cdots, J\}$. Each BS is equipped with an MEC server, which has limited resources for data processing, caching, and storage. The combination of each BS with its MEC server can be regarded as an edge node. Within the coverage area of the j-th BS, there are I_j mobile devices, denoted by a set $\mathcal{I}_j$, which have different delay-sensitive computation tasks to be processed. In this paper, we assume that each user has been already associated with a BS, which can be determined by some user association

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/TVT.2019.2904244, IEEE Transactions on Vehicular Technology

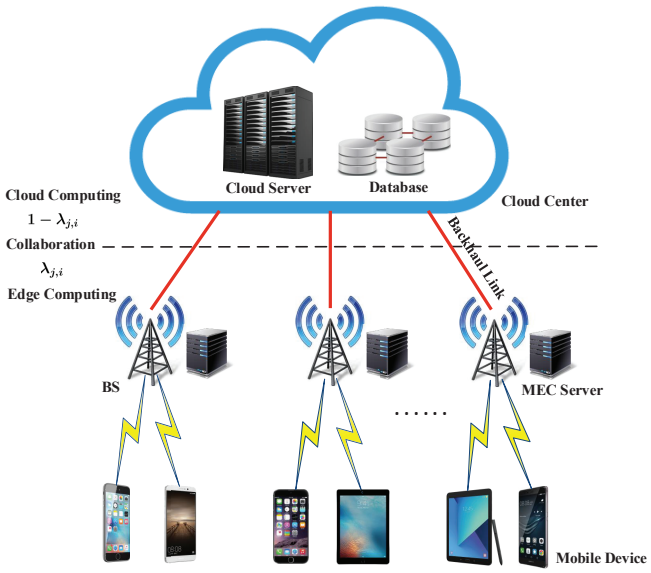


Fig. 1. Cloud-edge collaboration system.

strategies [30]–[32]. Moreover, each mobile device connects the corresponding BS through a wireless channel, and the edge nodes transmit data to the cloud server through different backhaul links.

In this system, we assume that each computation task can be processed at both the edge node and the cloud server [5], [13], [33]. Following the model in [5], we assume that all tasks have the same type and arrive simultaneously. Therefore we can use a two-field notation $A_{j,i} = \{L_{j,i}, C_{j,i}\}$ to denote the computation task of the i-th device served by the j-th edge node, where $L_{j,i}$ represents the input data-size (in bits) for processing the task and $C_{j,i}$ denotes the number of CPU cycles that are required to compute one-bit data of this task. Note that the two-field notation varies from different tasks and is known by the scheduler in advance. In addition, we define the computation capacity of the MEC server in the j-th edge node and the computation capacity of the cloud server as F_j^{e} and F^{c}, respectively, which are measured by the number of CPU cycles per second. The computation resources of each MEC server and the cloud server can be allocated to the mobile devices by the virtual machine technique [33].

B. Wireless Communication Model

The mobile devices adopt the TDMA method to share the system bandwidth. Specifically, each time frame is slotted and we denote the length of time-slot that is allocated to the i-th device served by the j-th edge node as $\tau_{j,i}$ ($\tau_{j,i} \geq 0$). The wireless channel between each mobile device and its connected BS is modeled as independent and identically distributed (i.i.d) Rayleigh variable. We assume that the perfect channel state information (CSI) can be estimated by the corresponding edge node and the cloud server so that the centralized scheduling algorithm can be implemented. We further denote the channel power gain from the i-th mobile device to the connected j-th BS in the n-th time-slot as $h_{j,i}^n$. In addition, let $p_{j,i}$ be the transmission power of the i-th device served by the

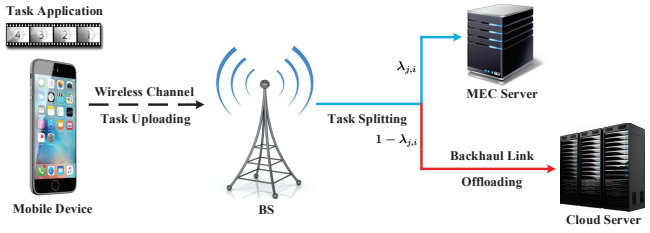


Fig. 2. The detailed procedures for task processing.

j-th edge node, which is assumed to be a constant in the whole transmission duration. Then, according to the Shannon formula, the achievable data rate (in bit/s) for each device in the n-th time-slot can be expressed as

$$r_{j,i}^n = B \log_2 \left(1 + \frac{p_{j,i} h_{j,i}^n}{\sigma_j}\right), \tag{1}$$

where B is the system bandwidth and σ_j is the noise power at the j-th edge node.

In this work, we assume that the wireless channel is orthogonally shared among different BSs. In this way, the problem can be properly analyzed and more insightful results can be obtained. In the future works, we can extend our analysis into more complicated non-orthogonal resource sharing scenarios where non-convex optimization tools could be applied.

C. Delay Analysis

In our model, we assume that each mobile device does not process the task directly due to its limited computation capacity. Alternatively, the tasks will be first offloaded to the corresponding edge node. Then, each edge node will determine whether the tasks should be processed itself, or be processed by the edge node and the cloud server collaboratively. In the latter case, the edge node should decide the proportion of each task that needs to be processed at the cloud server as well. Fig. 2 depicts the detailed operations of the cloud-edge collaboration system. From this figure, each device processes its task in the following five steps.

- The mobile device directly offloads the whole delay-sensitive task to the connected edge node through a wireless channel, without any local computing.
- The MEC server located in each edge node splits the received task into two parts with one part remained at the MEC server while the other part offloaded to the cloud server.
- The MEC server allocates its available computation resource to each serving device for edge computing, and simultaneously offloads part of data to the cloud server through a backhaul link.
- The cloud server allocates its total computation resource to all mobile devices for parallel cloud computing.
- The computation results are finally gathered by each edge node and then transmitted back to each mobile device.

It should be noted that the time to split a task is very small as compared with the corresponding computation and communication delay, and therefore can be neglected. Moreover, the size of computation result is also small enough so that the

download delay can be ignored, which corresponds to many practical computing scenarios, such as face recognition, virus detection, and video analysis [34]. To this end, there exist four kinds of delay in the whole process as illustrated in the following.

1) Transmission delay of mobile device: As mentioned before, each computation task should be directly transmitted to the connected BS through a wireless channel without any local computing. However, the channel power gain is random so that the accurate transmission delay is hard to calculate. Fortunately, the instantaneous transmission delay is typically in the timescale of millisecond due to the channel dynamics and short time-slot duration. On the other hand, the overall end-to-end delay of current mobile applications, such as online gaming, video conferencing, and 3D modelling, is usually in the range of tens or hundreds of milliseconds owing to the scarce communication bandwidth, limited computation capacity (CPU rather than GPU), and heavy computation workload. As a result, the overall execution process for each task will experience multiple time-slots so that it is reasonable to use the average transmission delay instead of the instantaneous transmission delay from the long timescale perspective. According to [35], we can evaluate the average transmission delay of the i-th device served by the j-th edge node in the following lemma.

Lemma 1: The average transmission delay for the i-th device to offload its computation task to the connected j-th BS is given by

$$t_{j,i}^{\text{tran,d}} = \frac{L_{j,i}T}{R_{j,i}\tau_{j,i}}, \tag{2}$$

where T represents the length of one TDMA frame, and $R_{j,i} = \mathbb{E}_{\boldsymbol{h}}\{r_{j,i}\}$ is the expected channel capacity, which can be realized through channel estimation for each device. Note that in this paper, we assume a fixed TDMA scheduling, i.e., the time-slot resource $\tau_{j,i}$ for each device is fixed in each time frame. In this case, the time duration for the i-th device to transmit data is $\tau_{j,i}$ while the other $T-\tau_{j,i}$ is its scheduling and waiting time. Therefore, the scheduling and waiting delay have been included in the transmission delay.

Proof: Please refer to Appendix A. ■

2) Computation delay of edge node: After receiving the whole data of each task from the mobile device, the MEC server will immediately launch the offloading strategy and split each task into two parts, with one part for edge computing and the other part for cloud computing. For general analysis, we assume that each computation task can be arbitrarily split without considering the inherent content, which corresponds to the scenarios like video compression and speech recognition [5], [13], [36]. Denote $\lambda_{j,i} \in [0,1]$ as the task splitting ratio, which accounts for the data proportion that is remained at the MEC server [5]. Then the number of CPU cycles needed for successfully processing the edge computing part of data can be expressed as $\lambda_{j,i}L_{j,i}C_{j,i}$. Following the computation model in [8], we define the computation resource that the j-th edge node allocates to the i-th device as $f_{j,i}^{\text{e}}$ (in CPU cycle/s). Therefore, the computation delay for processing the edge computing part of data at the MEC server can be expressed as

$$t_{j,i}^{\text{comp,e}} = \frac{\lambda_{j,i}L_{j,i}C_{j,i}}{f_{j,i}^{\text{e}}}. \tag{3}$$

3) Transmission delay of edge node: For each edge node, the communication module (transceiver) and the computation module (CPU/GPU) are generally separated. Therefore, the computation of the edge-computing part data can be in parallel with the transmission of the cloud-computing part data. Moreover, all edge nodes connect with the cloud server through different backhaul links, which are usually equipped with high bandwidth [38]. In fact, the backhaul link is shared by users and therefore its delay is very difficult to model due to the randomness of packet arrival, multi-user scheduling, complicated routing algorithm, and other factors [39]. To highlight our main contribution, i.e., the optimal collaboration policy for edge computing and cloud computing, we assume that the resource scheduling policy and routing algorithm are both fixed. Accordingly, we can denote W_j as the per-device backhaul communication capacity for each device associated with the j-th edge node. Therefore, similar to the average transmission delay in (2), the average backhaul transmission delay is proportional to the size of data to be offloaded, as

$$t_{j,i}^{\text{tran,e}} = \frac{(1-\lambda_{j,i})L_{j,i}}{W_j}, \tag{4}$$

where W_j^{-1} can be interpreted as the required time for the backhaul link to transmit one-bit data [39]–[41].

4) Computation delay of cloud server: When receiving the data from an edge node, the cloud server will allocate the available computation resource to each task for parallel computing. The number of CPU cycles for processing each cloud computing part of data can be calculated as $(1-\lambda_{j,i})L_{j,i}C_{j,i}$. Similar to the resource allocation method at the MEC servers, we denote $f_{j,i}^{\text{c}}$ (in CPU cycle/s) as the cloud computation resource that is allocated to the i-th device served by the j-th edge node. Therefore, the computation delay for processing the cloud computing part of data at the cloud server is given by

$$t_{j,i}^{\text{comp,c}} = \frac{(1-\lambda_{j,i})L_{j,i}C_{j,i}}{f_{j,i}^{\text{c}}}. \tag{5}$$

D. Problem Formulation

In the above, we have analyzed the four kinds of delay consisted in the cloud-edge collaboration system, i.e., the transmission delay of each mobile device, $t_{j,i}^{\text{tran,d}}$, the computation delay at each edge node, $t_{j,i}^{\text{comp,e}}$, the transmission delay between edge node and cloud server, $t_{j,i}^{\text{tran,e}}$, and the computation delay in the cloud server, $t_{j,i}^{\text{comp,c}}$. To determine the overall delay of each mobile device, we shall first give some reasonable assumptions as follows.

- Since task splitting operation depends on the specific parameters of each task, such as its data-size and computation workload, the MEC server distributed in each edge node cannot split a task until receiving the whole data of the task to better perform the task splitting strategy and guarantee the accuracy of the splitting results, according to [11] and [33].

- In practical systems, task computing may rely on the specific data structure as well as the correlation between adjacent data, such as video analysis in multimedia systems. To ensure the reliability of the computation result, the cloud server cannot start processing a task until the transmission between the edge node and the cloud server ends.

Based on the above assumptions, the overall delay of the i-th device served by the j-th edge node can be expressed as

$$T_{j,i} = t_{j,i}^{\text{tran,d}} + \max\left\{t_{j,i}^{\text{comp,e}}, t_{j,i}^{\text{tran,e}} + t_{j,i}^{\text{comp,c}}\right\}. \quad (6)$$

In this paper, we aim at minimizing the system delay of all mobile devices. We take the weighted-sum overall delay of all devices as evaluation criteria, which represents the average delay performance of the whole system. Moreover, we assign a positive weight factor $\beta_{j,i} \in (0,1)$ to each device, which stands for the fairness among mobile devices and satisfies $\sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i} = 1$. [1] The value of each weight factor, representing the level of importance, is determined according to the service priority of each mobile user. To this end, the optimization problem can be formulated as

$$\mathcal{P}_1: \min_{\substack{\{\tau_{j,i},\lambda_{j,i},\\ f_{j,i}^{\text{c}},f_{j,i}^{\text{e}}\}}} \sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i}T_{j,i}, \quad (7a)$$

$$\text{s.t.} \quad \sum_{j=1}^{J}\sum_{i=1}^{I_j}\tau_{j,i} \le T,\ \tau_{j,i} \ge 0, \quad (7b)$$

$$\sum_{j=1}^{J}\sum_{i=1}^{I_j} f_{j,i}^{\text{c}} \le F^{\text{c}},\ f_{j,i}^{\text{c}} \ge 0, \quad (7c)$$

$$\sum_{i=1}^{I_j} f_{j,i}^{\text{e}} \le F_j^{\text{e}},\ f_{j,i}^{\text{e}} \ge 0,\ \forall j \in \mathcal{J}, \quad (7d)$$

$$0 \le \lambda_{j,i} \le 1,\ \forall i \in \mathcal{I}_j,\ \forall j \in \mathcal{J}, \quad (7e)$$

where (7b) is the overall communication resource constraint of all mobile devices, (7c) and (7d) imply that the allocated computation resources for mobile devices should not exceed the maximum available resources provided by the cloud server and each edge node, respectively. The optimization variables include the time-slot allocation $\{\tau_{j,i}\}$, the computation resource allocation $\left\{f_{j,i}^{\text{c}}, f_{j,i}^{\text{e}}\right\}$, and the task splitting ratio $\{\lambda_{j,i}\}$.

The above problem is not easy to solve due to the piecewise expression of $T_{j,i}$ in (6). In the next, we will first decompose $\mathcal{P}_1$ into two equivalent subproblems and then develop the optimal solutions. Through the proposed solutions, some insightful results will be also highlighted.

III. Communication Resource Allocation

In this section, we will first decompose the optimization problem $\mathcal{P}_1$ into two equivalent subproblems. After that, the closed-form expression for the optimal communication resource allocation policy will be devised, which also facilitates the optimal task splitting strategy and computation resource allocation in Sections IV and V, respectively.

A. Problem Decomposition

To solve $\mathcal{P}_1$, we first analyze the structural characteristics of the overall delay of each device. From (2), the average transmission delay of each mobile device, $t_{j,i}^{\text{tran,d}}$, is determined only by the length of time-slot, $\tau_{j,i}$, and is independent of other optimization variables. Meanwhile, the transmission delay from each edge node to the cloud server, $t_{j,i}^{\text{tran,e}}$, the computation delay at each edge node, $t_{j,i}^{\text{comp,e}}$, and the computation delay in the cloud server, $t_{j,i}^{\text{comp,c}}$ are all independent of $\tau_{j,i}$. Furthermore, the constraint in (7b) does not couple with other constraints. With this consideration, the cloud-edge collaboration system can be divided into two independent subsystems, corresponding to the transmission system between mobile devices and edge nodes and the collaborative computation system between edge nodes and the cloud server.

Accordingly, $\mathcal{P}_1$ can be equivalently decomposed into two subproblems. The first one is to minimize the weighted-sum transmission delay between each mobile device and the connected BS, and can be formulated as

$$\mathcal{P}_2: \min_{\{\tau_{j,i}\}} \sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i}t_{j,i}^{\text{tran,d}}, \quad (8)$$

$$\text{s.t.} \quad (7b),$$

which will be investigated in this section. The other one is to minimize the weighted-sum computation delay of each edge node and the cloud server, and can be expressed as

$$\mathcal{P}_3: \min_{\{f_{j,i}^{\text{c}},f_{j,i}^{\text{e}},\lambda_{j,i}\}} \sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i}t_{j,i}^{\text{comp}}, \quad (9)$$

$$\text{s.t.} \quad (7c),\ (7d),\ \text{and}\ (7e),$$

where $t_{j,i}^{\text{comp}} = \max\left\{t_{j,i}^{\text{comp,e}}, t_{j,i}^{\text{tran,e}} + t_{j,i}^{\text{comp,c}}\right\}$ is the computation delay of the i-th mobile device served by the j-th edge node. The problem $\mathcal{P}_3$ will be discussed in Sections IV and V later.

It should be noted that the decomposing procedure preserves the optimality due to the independence of communication and computation resource allocations.

B. Optimal Communication Resource Allocation

The communication resource allocation is associated with the weighted-sum transmission delay between the mobile devices and the corresponding edge nodes. Therefore, we can derive the optimal communication resource allocation solution by solving $\mathcal{P}_2$. The detailed result is summarized in the following theorem.

Theorem 1: For the optimal communication resource allocation policy of the cloud-edge collaboration system, the time-slot allocated to the i-th device served by the j-th edge node

[1] In practical systems, the weighted min-max delay of all devices is another evaluation criteria to be optimized, whereas its solution is identical to the weighted-sum problem by properly adjusting the weight value according to [42].

can be expressed as

$$\tau_{j,i}^{*}=\frac{\sqrt{\frac{\beta_{j,i}L_{j,i}}{R_{j,i}}}}{\sum_{j=1}^{J}\sum_{i=1}^{I_j}\sqrt{\frac{\beta_{j,i}L_{j,i}}{R_{j,i}}}}T. \tag{10}$$

Proof: Please refer to Appendix B. ■

Remark 1: Theorem 1 reveals that the optimal time-slot allocated to the i-th device served by the j-th edge node is determined by the corresponding weight factor $\beta_{j,i}$, the input data-size $L_{j,i}$, and the wireless channel capacity $R_{j,i}$. The weight factor $\beta_{j,i}$ indicates the level of importance of each device in the system and represents the priority of this device. The data-size $L_{j,i}$ stands for the workload of transmitting the task to the connected BS and the channel capacity $R_{j,i}$ accounts for the channel condition of the wireless link. As the size of the input-data grows or the channel condition deteriorates, the transmission delay of this device will correspondingly increase, leading to more time-slots to be allocated.

IV. Optimal Task Splitting Strategy

We shall note that the objective function in (9) is so complicated that we cannot solve $\mathcal{P}_3$ directly. To this end, we will first determine the optimal task splitting ratio $\lambda_{j,i}^{*}$ while keeping $f_{j,i}^{\mathrm{e}}$ and $f_{j,i}^{\mathrm{c}}$ fixed in this section.

For ease of notation, we first define two significative parameters for each mobile device.

1) The **normalized backhaul communication capacity** is defined as the ratio between the backhaul communication capacity and the edge computation capacity, i.e., $\eta_{j,i}=\frac{C_{j,i}W_j}{f_{j,i}^{\mathrm{e}}}$.

2) The **normalized cloud computation capacity** is defined as the ratio between the cloud computation capacity and the edge computation capacity, i.e., $\gamma_{j,i}=\frac{f_{j,i}^{\mathrm{c}}}{f_{j,i}^{\mathrm{e}}}$.

On the basis of the above definitions, we can derive the optimal task splitting strategy, as presented in Theorem 2.

Theorem 2: The optimal task splitting strategy, $\left\{\lambda_{j,i}^{*}\right\}$, for the cloud-edge collaboration system can be expressed as

$$\lambda_{j,i}^{*}=\frac{\eta_{j,i}+\gamma_{j,i}}{\eta_{j,i}+\gamma_{j,i}+\eta_{j,i}\gamma_{j,i}}. \tag{11}$$

Proof: Please refer to Appendix C. ■

Remark 2: Theorem 2 reveals that the optimal task splitting strategy depends only on the two ratios: the normalized backhaul communication capacity and the normalized cloud computation capacity. Furthermore, the optimal task splitting strategy is determined by the harmonic average of these two ratios. It can be easily verified that the proportion of task data processed at the edge node will correspondingly decrease as $\eta_{j,i}$ or $\gamma_{j,i}$ increases. Therefore, when a mobile device is allocated with little edge computation resource while having adequate cloud computation resource, it should offload more task data to the cloud server. On the contrary, more task data should be processed at the edge node if the cloud computation resource is very scarce. It can be easily explained that we are willing to assign more task data to more powerful server, which will significantly reduce the overall delay of the mobile device.

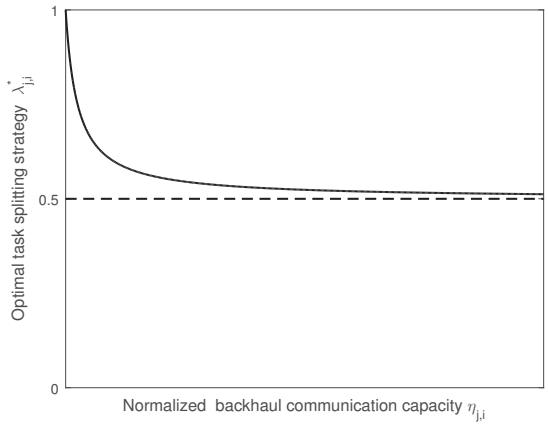


Fig. 3. Illustration of the optimal task splitting strategy with $\eta_{j,i}$ when $\gamma_{j,i}=1$.

Fig. 3 depicts the relation between $\lambda_{j,i}^{*}$ and $\eta_{j,i}$ while keeping $\gamma_{j,i}=1$. From the figure, more task data should be offloaded to the cloud server as the normalized backhaul communication capacity increases for a given normalized cloud computation capacity. Note that the limit of the splitting ratio $\lambda_{j,i}^{*}$ is 0.5 since the computation capacity of the edge node equals to that of the cloud server in this case. The relation between $\lambda_{j,i}^{*}$ and $\gamma_{j,i}$ for fixing $\eta_{j,i}$ is similar to the curve in Fig. 3 since $\gamma_{j,i}$ and $\eta_{j,i}$ are symmetric in (11).

Now we discuss some inherent insights of the results in Theorem 2. We consider four different systems as follows.

Communication-limited system: For the communication-limited system, the communication capacities between the edge nodes and the cloud server are insufficient while the computation resources are sufficient. In this case, we have $\eta_{j,i}\ll\gamma_{j,i},\forall j,i$. It happens when the number of mobile devices served by the j-th edge node is tremendous while the communication capacity of the backhaul link is limited. In this case, we have

$$\lambda_{j,i}^{*(1)}=\lim_{\frac{\eta_{j,i}}{\gamma_{j,i}}\to 0}\lambda_{j,i}^{*}=\frac{1}{1+\eta_{j,i}}, \tag{12}$$

which means that the optimal task splitting strategy in this scenario is determined only by the normalized backhaul communication capacity and will not be influenced by the normalized cloud computation capacity. In this scenario, the communication resource is the main bottleneck of reducing the end-to-end delay of each device so that the relative size of the edge computation capacity and the cloud computation capacity is not significant anymore. In the more special case that $\eta_{j,i}\to 0$, the optimal task splitting ratio $\lambda_{j,i}^{*}$ achieves 1, which indicates that the whole task should be processed at the edge node without offloading to the cloud server. This agrees with our intuition that offloading tasks to the cloud server will intensify the network congestion and cause long transmission latency when the communication capacity of the backhaul link is insufficient.

Computation-limited system: For the computation-limited system, the computation capacities of the edge nodes and the cloud server are limited while the communication capacities of the backhaul links are adequate. In this case, we have $\eta_{j,i} \gg \gamma_{j,i}, \forall j, i$ and

$$\lambda_{j,i}^{*(2)} = \lim_{\frac{\eta_{j,i}}{\gamma_{j,i}} \to \infty} \lambda_{j,i}^* = \frac{1}{1+\gamma_{j,i}}. \tag{13}$$

From this result, the optimal task splitting ratio is determined only by the normalized cloud computation capacity. In this scenario, the backhaul communication capacity is relatively sufficient so that the computation delay dominates the overall delay of each device. Notice that in this case, the edge node and the cloud server can be regarded as a whole, and thus the task should be proportionally split according to the ratio of their computation capacities, i.e., $\gamma_{j,i}$. To be more specific, if the computation capacity of the edge node is larger than that of the cloud server, i.e., $\gamma_{j,i} < 1$, more data should be processed at the edge node. Otherwise, more data should be offloaded to the cloud server. In the special case of $\gamma_{j,i} = 1$, the data should be equally split between the edge node and the cloud server.

Edge-dominated system: For the edge-dominated system, the allocated computation resource of the edge node is much greater than that of the cloud server, i.e., $f_{j,i}^{\mathrm{e}} \gg f_{j,i}^{\mathrm{c}}$. Therefore, the normalized cloud computation capacity $\gamma_{j,i} \to 0$, which corresponds to the large-scale small-cell cellular networks where the cloud server serves a lot of edge nodes. In this system, we can obtain the limit of the optimal task splitting ratio, as

$$\lambda_{j,i}^{*(3)} = \lim_{\gamma_{j,i} \to 0} \lambda_{j,i}^* = 1, \tag{14}$$

which indicates that the whole task should be processed at the edge node without offloading to the cloud server. The reason is that, when the cloud computation capacity is much smaller than that of the edge node, offloading task data to the cloud server may incur extra transmission delay and cause longer computation delay as well. Even if the normalized backhaul communication capacity is large enough, the cloud computation delay still dominates the overall delay. Therefore, the whole task should be processed at the edge node only.

Cloud-dominated system: For the cloud-dominated system, the cloud server is equipped with enormous computation capacity while the computation capacity of each edge node is insufficient. In this case, the normalized cloud computation capacity $\gamma_{j,i} \to \infty$. This kind of system corresponds to the scenario with powerful cloud servers and weak edge computation capacity. By inserting $\gamma_{j,i} \to \infty$ into (11), we have

$$\lambda_{j,i}^{*(4)} = \lim_{\gamma_{j,i} \to \infty} \lambda_{j,i}^* = \frac{1}{1+\eta_{j,i}}, \tag{15}$$

which indicates that the optimal task splitting strategy in this system is determined only by the normalized backhaul communication capacity. If the backhaul communication capacity W_j becomes larger or the edge computation capacity $f_{j,i}^{\mathrm{e}}$ gets smaller, the optimal task splitting ratio will correspondingly decrease. It is because the cloud computation delay is negligible comparing with the edge computation delay when $\gamma_{j,i} \to \infty$. Therefore, when the normalized backhaul communication capacity $\eta_{j,i} < 1$, the backhaul transmission delay will dominate the overall delay of this mobile device, which leads to a larger proportion of task data for edge processing, i.e., $\lambda_{j,i}^* > \frac{1}{2}$. Otherwise, when $\eta_{j,i} > 1$, more task data should be offloaded to the cloud server for processing because of the larger backhaul communication capacity.

V. Computation Resource Allocation

In Sections III and IV, we have discussed the optimal communication resource allocation and the optimal task splitting strategy, respectively. In this section, we will investigate the computation resource allocation.

A. The Optimal Solution

First, by applying the detailed expression of $\lambda_{j,i}^*$ into (9), the task splitting ratio can be eliminated and the computation delay $t_{j,i}^{\mathrm{comp}}$ can be rewritten as

$$\widehat{t_{j,i}^{\mathrm{comp}}} = \frac{C_{j,i} f_{j,i}^{\mathrm{c}} + C_{j,i}^2 W_j}{f_{j,i}^{\mathrm{e}} f_{j,i}^{\mathrm{c}} + C_{j,i} W_j \left(f_{j,i}^{\mathrm{e}} + f_{j,i}^{\mathrm{c}}\right)} L_{j,i}. \tag{16}$$

Then $\mathcal{P}_3$ can be equivalently converted into the following problem, as

$$\mathcal{P}_4: \quad \min_{\{f_{j,i}^{\mathrm{c}}, f_{j,i}^{\mathrm{e}}\}} \sum_{j=1}^{J} \sum_{i=1}^{I_j} \beta_{j,i} \widehat{t_{j,i}^{\mathrm{comp}}}, \tag{17}$$

$$\text{s.t.} \quad \text{(7c), (7d).}$$

Lemma 2: $\mathcal{P}_4$ is a convex optimization problem.

Proof: Please refer to Appendix D. ∎

Based on Lemma 2, $\mathcal{P}_4$ can be solved by the Karush-Kuhn-Tucker (KKT) conditions and we can derive the optimal computation resource allocation policy, as presented in Theorem 3.

Theorem 3: The optimal computation resource allocation policy for the cloud-edge collaboration system is given by

$$\begin{cases} f_{j,i}^{\mathrm{e}*} = \left[\sqrt{\dfrac{\beta_{j,i} L_{j,i} C_{j,i}}{\mu_j^*}} - \left(1 - \sqrt{\dfrac{\theta^*}{\mu_j^*}}\right) C_{j,i} W_j\right]^+, \\ f_{j,i}^{\mathrm{c}*} = \left(\sqrt{\dfrac{\mu_j^*}{\theta^*}} - 1\right)^+ C_{j,i} W_j, \end{cases} \tag{18}$$

where $(x)^+ = \max\{x, 0\}$, $\mu_j^*, \forall j \in \mathcal{J}$ and θ^* are the optimal values of Lagrange multipliers that satisfy the computation resource constraints $\sum_{i=1}^{I_j} f_{j,i}^{\mathrm{e}*} = F_j^{\mathrm{e}}$ and $\sum_{j=1}^{J} \sum_{i=1}^{I_j} f_{j,i}^{\mathrm{c}*} = F^{\mathrm{c}}$ simultaneously.

Proof: Please refer to Appendix E. ∎

Remark 3: Theorem 3 reveals that the computation resources of the edge node and the cloud server that are allocated to the i-th mobile device served by the j-th edge node are mainly determined by the required CPU cycles for processing one-bit input-data, $C_{j,i}$, and the transmission capacity of the corresponding backhaul link, W_j. As $C_{j,i}$ increases or W_j decreases, more edge and cloud computation resources should be allocated to this device for a smaller weighted-sum delay.

B. A Necessary Condition

From Theorem 3, the allocated cloud computation resource $f_{j,i}^{c*}$ will become zero if $\mu_j^* \leq \theta^*$, which indicates that all the tasks associated with this edge node will be processed at the MEC server without offloading to the cloud server. In the following, we will present a necessary condition for each edge node to not offload tasks to the cloud server.

Corollary 1: If there exists an edge node $k \in \mathcal{J}$ that the associated tasks are all processed itself, the computation capacity of this edge node must satisfy

$$F_k^{\mathrm{e}} \geq \rho F^{\mathrm{c}} \frac{\sum_{i=1}^{I_k} \sqrt{\beta_{k,i} L_{k,i} C_{k,i}}}{\sum_{j=1}^{J} \sum_{i=1}^{I_j} \sqrt{\beta_{j,i} L_{j,i} C_{j,i}}}, \tag{19}$$

where $\rho = \min_{j=1}^{J} \left\{ \frac{F_j^{\mathrm{e}}}{\sum_{i=1}^{I_j} C_{j,i} W_j} \right\}$ is a constant coefficient that can be interpreted as the minimum ratio between the edge computation capacity and the backhaul communication capacity among all edge nodes.

Proof: Please refer to Appendix F. ∎

It should be noted that Corollary 1 is a necessary but not a sufficient condition. Nevertheless, we can still use this corollary to judge whether all tasks within an edge node should not be offloaded to the cloud server. From Corollary 1, we can further define a new parameter as

$$Q_j = \frac{F_j^{\mathrm{e}}}{\sum_{i=1}^{I_j} \sqrt{\beta_{j,i} L_{j,i} C_{j,i}}}, \tag{20}$$

which can be interpreted as the *relative computation capacity* of each edge node. It can be seen that Q_j is mainly determined by the computation capacity of the edge node and the total task workload of the connected devices. For each edge node, when its computation capacity, F_j^{e}, is large, or the computation workload of its associated mobile devices is low, its Q_j will be large. Moreover, the edge node with a larger Q_j is more likely to fulfill the condition in Corollary 1. Thus, the Q_j defined in (20) provides a significant reference for the design of collaborative computing between edge node and cloud server.

VI. Simulation Results

In this section, we will present numerical results to confirm our theoretical analysis and demonstrate the performance of the proposed algorithms. In the simulation, the radius of each edge node is 500 m. Within the coverage area, there are several mobile devices randomly distributed and served by the corresponding BS through a wireless channel. All mobile devices have the same weight factors, therefore the system delay accounts for the average delay of all devices. The channel gains between mobile devices and edge nodes are generated according to i.i.d. Rayleigh random variables with unit variance. For each computation task, the input data-size and the workload for computing one-bit data follow the uniform distribution with $L_{j,i} \in$ [0.1, 0.5] Mbits and $C_{j,i} \in$ [500, 1500] CPU cycle/bit, respectively. The detailed simulation parameters are listed in Table I unless otherwise stated.

We first compare the performance of our cloud-edge collaboration scheme with other three baseline schemes: the *edge only scheme*, where all the tasks are offloaded to the connected MEC servers for edge computing only without any cloud computing; the *cloud only scheme*, where all tasks are offloaded to the cloud server for parallel cloud computing; and the *fixed cloud-edge scheme*, where half of each task is processed at the edge node while the other half is offloaded for cloud computing. We name our proposed scheme as the *collaborative cloud-edge scheme* in the sequel.

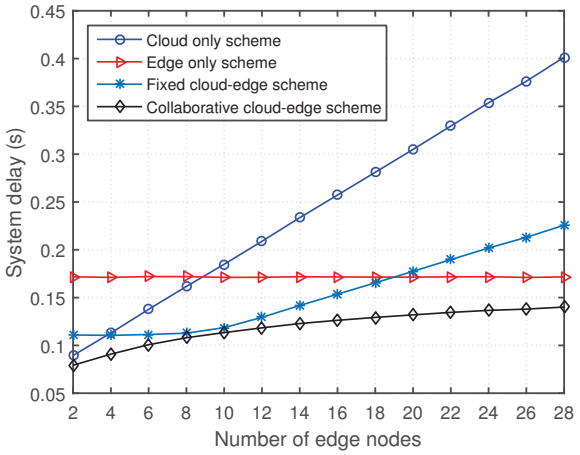
TABLE I
System Parameters

Parameters	Settings
Cell radius, r	500 m
System bandwidth, B	10 MHz
Noise power density, σ_j	-174 dBm/Hz
Pass loss between user and BS	$128.1 + 37.6 \log_{10}(d(\text{km}))$
Length of time frame, T	100 ms
Transmission power, $p_{j,i}$	24 dBm
Input data-size, $L_{j,i}$	$[0.1, 0.5]$ Mbits
Workload for one-bit data, $C_{j,i}$	$[500, 1500]$ CPU cycle/bit
Backhaul link capacity, W_j	$[5, 50]$ Mbps
Edge computation capacity, F_j^{e}	$\left[2 \times 10^{10}, 8 \times 10^{10}\right]$ CPU cycle/s
Cloud computation capacity, F^{c}	$\left[1.5 \times 10^{11}, 6 \times 10^{11}\right]$ CPU cycle/s

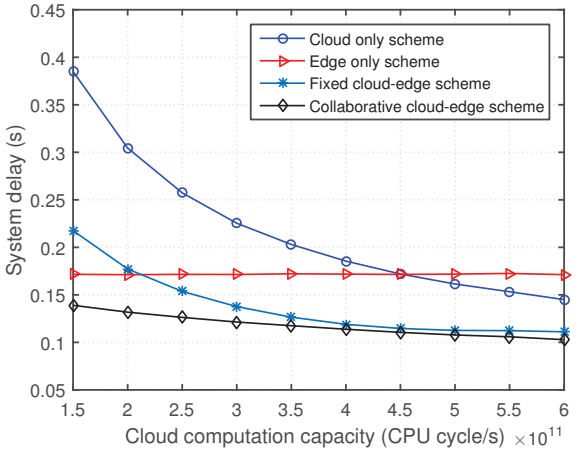
Fig. 4(a) depicts the average system delay versus the number of edge nodes in the four schemes where we fix the total cloud computation capacity $F^{\mathrm{c}} = 5 \times 10^{11}$ CPU cycles per second and assume that each edge node serves 20 mobile devices. From the figure, the system delays of the *cloud only scheme*, *fixed cloud-edge scheme*, and *collaborative cloud-edge scheme* increase with the number of edge nodes because of the limited cloud computation resource. Meanwhile, the system delay of the *edge only scheme* does not change with the number of edge nodes since it does not utilize the cloud computing. Moreover, when the number of edge nodes is small, the *cloud only scheme* performs better than both the *edge only scheme* and the *fixed cloud-edge scheme*. It is because, when the number of edge nodes is small, the cloud computation resource allocated to each device is always larger than the corresponding edge computation resource, i.e., $\gamma_{j,i} > 1$. Therefore, we should offload more data to the cloud server for the sake of delay minimization. However, as the number of edge nodes increases, the *edge only scheme* will perform better than the *cloud only scheme* due to the limited cloud computation resource. Especially when the number of edge nodes is very large (more than 19 in our simulation), the *edge only scheme* even achieves a better performance than the *fixed cloud-edge scheme*, which indicates that more data should be assigned for edge computing than for cloud computing. In all cases, the *collaborative cloud-edge scheme* always has the best performance among all schemes since it can utilize the computation capacity of each edge node and the cloud server optimally.

Fig. 4(b) shows the average system delay versus the computation capacity of the cloud server, where the number of edge nodes is fixed to be 8. When the computation capacity of the cloud server increases, the system delay of the *cloud only scheme* will decrease significantly. In this situation, offloading

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/TVT.2019.2904244, IEEE Transactions on Vehicular Technology

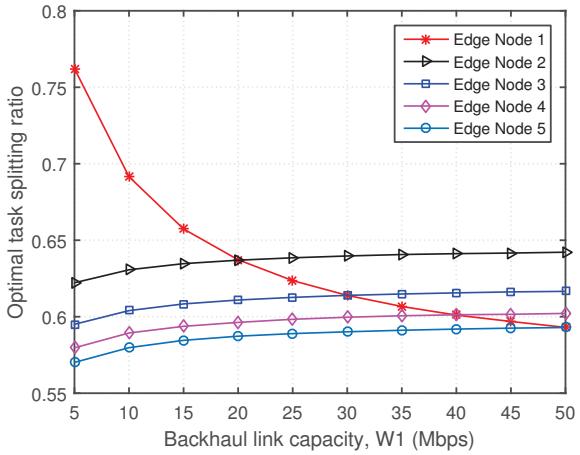


(a) Average system delay versus the number of edge nodes.

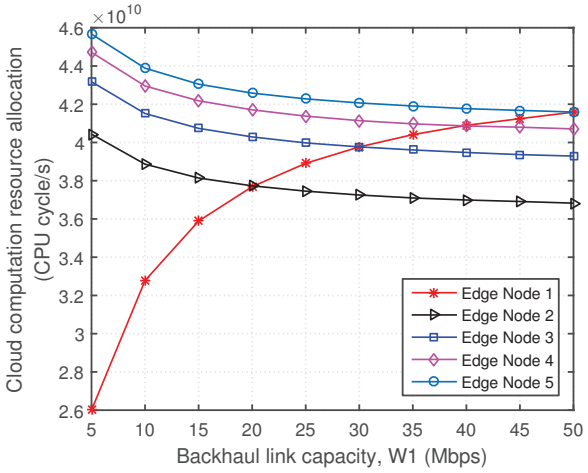


(b) Average system delay versus the cloud computation capacity.

Fig. 4. Average system delay of different schemes.



(a) Task splitting ratio.



(b) Computation resource allocation.

Fig. 5. Optimal task splitting ratio and computation resource allocation with different backhaul link capacities.

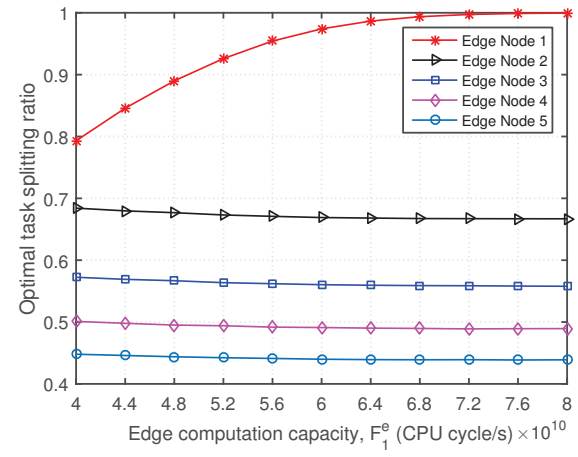
tasks to the cloud server will achieve less delay than processing the tasks at the MEC servers, which eventually reduces the whole system delay. On the contrary, when the computation capacity of the cloud server becomes smaller, the *edge only scheme* will perform better than the *cloud only scheme*. Similar to Fig. 4(a), the *collaborative cloud-edge scheme* always achieves the best performance among all schemes.

We now analyze the results of the optimal task splitting ratio and cloud computation resource allocation with different system parameters. In this test, we assume that the system consists of 5 edge nodes, each has 10 mobile devices. For simplicity, we further assume that the mobile devices associated with the same edge node have the same simulation parameters, so that their task splitting ratios and cloud computation resource allocations are identical. Under this configuration, we then change the parameters of the first edge node while fixing the parameters of other four edge nodes to show the different results of the task splitting ratio and cloud computation resource allocation.

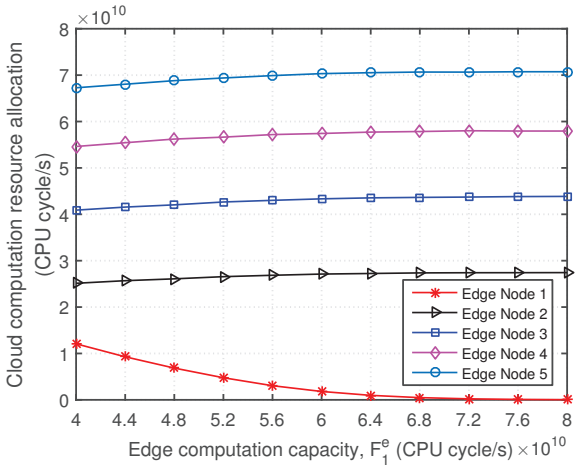
Fig. 5 depicts the optimal task splitting ratio and cloud computation resource allocation with different backhaul link capacities of the first edge node where the backhaul capacities of other four edge nodes are 20 Mbps, 30 Mbps, 40 Mbps, and 50 Mbps, respectively. The cloud computation capacity is assumed to be 2×10^{11} CPU cycles per second. From Fig. 5(a) and Fig. 5(b), the proportion of data processed in the first edge node decreases with its backhaul link capacity, implying that more task data should be offloaded for cloud computing. Correspondingly, the cloud computation resource allocated to the first edge node will also increase. The reason can be explained as follows. When the backhaul link capacity increases, it will consume less transmission time for the corresponding edge node to offload data to the cloud server. Therefore, to achieve a better delay performance, more task data should be offloaded to the cloud server. On the contrary, the cloud computation resource allocated to other edge nodes will correspondingly decrease and more data should be processed at these edge nodes, leading to the increase of their task splitting ratios.

Fig. 6 illustrates the optimal task splitting ratio and cloud computation resource allocation with different computation capacities of the first edge node where the computation capacities of other four edge nodes are 4.6×10^{10}, 4.4×10^{10}, 4.2×10^{10}, and 4.0×10^{10} CPU cycles per second, respectively. From the

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/TVT.2019.2904244, IEEE Transactions on Vehicular Technology



(a) Task splitting ratio.



(b) Computation resource allocation.

Fig. 6. Optimal task splitting ratio and computation resource allocation with different edge computation capacities.

figure, the proportion of data processed at the first edge node increases with its computation capacity while the allocated cloud computation resource decreases. The results for the other four edge nodes are just the opposite. The reason is very intuitive as we should fully utilize the computation resource of the edge nodes with strong computation capacities while allocating more cloud computation resource to assist the edge nodes with relative weak computation capacities. It should be also emphasized that when the computation capacity of the first edge node exceeds a threshold, all the related tasks should be processed at the MEC server only, without offloading to the cloud server, which demonstrates the accuracy and applicability of Corollary 1.

VII. Conclusion

In this paper, we have investigated joint communication and computation resource allocation to minimize the weighted-sum delay of all devices in a cloud-edge collaboration system. A latency minimization problem with the constraints of communication and computation resources has been firstly formulated. To tackle it, we have decomposed the optimization problem into two subproblems. The first one is associated with the communication resource allocation whose optimal solution can be expressed in closed-form by using the Cauchy-Buniakowsky-Schwarz inequality. The other subproblem corresponds to the computation resource allocation between the edge nodes and the cloud server. We have found that the optimal task splitting strategy for each mobile device is determined by two significative parameters: the *normalized backhaul communication capacity* and the *normalized cloud computation capacity*. We have further highlighted some inherent insights of the optimal task splitting strategy by analyzing four special scenarios. Based on this, the optimal computation resource allocation can be finally derived by utilizing the KKT conditions. Our initial study here sheds light on the design of collaborative cloud and edge computing in future mobile computing systems.

In this work, we utilize the weighted-sum delay of all devices as the performance metric, which may not guarantee the latency requirement of each single user. Our analysis can be extended to more general scenarios where each user's latency requirement is considered as an additional constraint in $\mathcal{P}_1$. Moreover, to gain closed-form and insightful results, user mobility, asynchronous task arrival, base station cooperation, and server scheduling queuing are not investigated in our work. These issues, however, are important and interesting, which can be emphasized in the future work. Also, in our computation model, computing at local devices is not considered. Future works can collaborate local, edge, and cloud computation capacities for further performance enhancement.

Appendix A
Proof of Lemma 1

To prove Lemma 1, we shall first define $N_{j,i}^*$ as the least number of time-slots required by the i-th device to transmit its computation task to the connected j-th edge node, which is given by

$$N_{j,i}^* = \arg\min \left\{ N_{j,i} : \sum_{n=1}^{N_{j,i}} r_{j,i}^n \geq \frac{L_{j,i}}{\tau_{j,i}} \right\}. \tag{21}$$

Since the channel power gain $h_{j,i}^n$ is an i.i.d. and infinite real-valued random variable across different time-slots, the data rate $\{r_{j,i}^n\}$ can be regarded as a sequence of infinite real-valued random variables based on the definition of channel capacity in (1). Therefore, the slot number $N_{j,i}^*$ is also an i.i.d. random variable associated with $h_{j,i}^n$. Based on this, we can prove Lemma 1 as follows.

On the one hand, by applying the Wald's Equation in martingale theory [43], we have

$$\mathbb{E}_{\boldsymbol{h}} \left\{ \sum_{n=1}^{N_{j,i}^*} r_{j,i}^n \right\} = \mathbb{E}_{\boldsymbol{h}} \left\{ N_{j,i}^* \right\} \mathbb{E}_{\boldsymbol{h}} \left\{ r_{j,i} \right\}, \tag{22}$$

where $\mathbb{E}_{\boldsymbol{h}} \{r_{j,i}\}$ is the expectation of the channel capacity over the channel power gain $\boldsymbol{h}$. On the other hand, according to equation (21), we have

$$\mathbb{E}_{\boldsymbol{h}} \left\{ \sum_{n=1}^{N_{j,i}^*} r_{j,i}^n \right\} = \mathbb{E}_{\boldsymbol{h}} \left\{ \frac{L_{j,i}}{\tau_{j,i}} \right\} = \frac{L_{j,i}}{\tau_{j,i}}, \tag{23}$$

where $\tau_{j,i}$ is the length of time-slot allocated to the i-th device served by the j-th edge node.

As we adopt the TDMA method, the accurate transmission delay can be characterized as $N^*_{j,i}T$, where T is the length of one TDMA frame. Then, by combining (22) with (23), the average transmission delay can be evaluated as

$$\begin{aligned} t^{\text{tran,d}}_{j,i} &= \mathbb{E}_{\boldsymbol{h}}\left\{N^*_{j,i}T\right\} = T\mathbb{E}_{\boldsymbol{h}}\left\{N^*_{j,i}\right\} \\ &= T\frac{\mathbb{E}_{\boldsymbol{h}}\left\{\sum_{n=1}^{N^*_{j,i}} r^n_{j,i}\right\}}{\mathbb{E}_{\boldsymbol{h}}\left\{r_{j,i}\right\}} = \frac{T}{\tau_{j,i}}\frac{L_{j,i}}{\mathbb{E}_{\boldsymbol{h}}\left\{r_{j,i}\right\}} = \frac{L_{j,i}T}{R_{j,i}\tau_{j,i}}, \end{aligned} \tag{24}$$

where $R_{j,i} = \mathbb{E}_{\boldsymbol{h}}\left\{r_{j,i}\right\}$. This ends the proof.

APPENDIX B
PROOF OF THEOREM 1

To prove Theorem 1, we first prove that the optimal solution $\{\tau^*_{j,i}\}$ to $\mathcal{P}_2$ satisfies $\sum_{j=1}^{J}\sum_{i=1}^{I_j}\tau^*_{j,i} = T$ by contradiction. Assume that there is another optimal solution $\{\tau^{*'}_{j,i}\}$ to $\mathcal{P}_2$ that satisfies $\sum_{j=1}^{J}\sum_{i=1}^{I_j}\tau^{*'}_{j,i} = T-\delta$, where $\delta \in (0,T)$ represents the remained amount of time-slot. Since the transmission delay for each device, i.e., $t^{\text{tran,d}}_{j,i} = \dfrac{L_{j,i}T}{R_{j,i}\tau_{j,i}}$ decreases with $\tau_{j,i}$. Therefore, if we assign the remained time-slot δ to any device, the corresponding transmission delay will decrease while the transmission delay of other devices remain unchange. As a result, the objective function of $\mathcal{P}_2$, i.e., $\sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i}t^{\text{tran,d}}_{j,i}$ will also decrease, which contradicts to the requirement for the optimal solution to $\mathcal{P}_2$. To this end, we can conclude that the optimal solution to $\mathcal{P}_2$ satisfies $\sum_{j=1}^{J}\sum_{i=1}^{I_j}\tau^*_{j,i} = T$.

Then, according to the Cauchy-Buniakowsky-Schwarz inequality, we have

$$\begin{aligned} \sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i}\frac{L_{j,i}T}{R_{j,i}\tau^*_{j,i}} &= \left(\sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i}\frac{L_{j,i}T}{R_{j,i}\tau^*_{j,i}}\right)\left(\sum_{j=1}^{J}\sum_{i=1}^{I_j}\frac{\tau^*_{j,i}}{T}\right) \\ &\geq \left(\sum_{j=1}^{J}\sum_{i=1}^{I_j}\sqrt{\beta_{j,i}\frac{L_{j,i}T}{R_{j,i}\tau^*_{j,i}}\frac{\tau^*_{j,i}}{T}}\right)^2 = \left(\sum_{j=1}^{J}\sum_{i=1}^{I_j}\sqrt{\frac{\beta_{j,i}L_{j,i}}{R_{j,i}}}\right)^2. \end{aligned} \tag{25}$$

The equality holds iff $\sqrt{\dfrac{\beta_{j,i}L_{j,i}}{R_{j,i}}}\Big/\dfrac{\tau^*_{j,i}}{T}$ are equal for all $i \in \mathcal{I}_j$, $j \in \mathcal{J}$. Assume that $\sqrt{\dfrac{\beta_{j,i}L_{j,i}}{R_{j,i}}}\Big/\dfrac{\tau^*_{j,i}}{T} = B$, then we have $\tau^*_{j,i} = \sqrt{\dfrac{\beta_{j,i}L_{j,i}}{B^2R_{j,i}}}T$. Inserting these results into the equality constraint $\sum_{j=1}^{J}\sum_{i=1}^{I_j}\tau^*_{j,i} = T$, we can derive that $B = \sum_{j=1}^{J}\sum_{i=1}^{I_j}\sqrt{\dfrac{\beta_{j,i}L_{j,i}}{R_{j,i}}}$. Consequently, the optimal solution $\tau^*_{j,i}$ can be expressed as

$$\tau^*_{j,i} = \frac{\sqrt{\dfrac{\beta_{j,i}L_{j,i}}{R_{j,i}}}}{\sum_{j=1}^{J}\sum_{i=1}^{I_j}\sqrt{\dfrac{\beta_{j,i}L_{j,i}}{R_{j,i}}}}T. \tag{26}$$

This completes the proof.

APPENDIX C
PROOF OF THEOREM 2

We can prove Theorem 2 by analyzing the monotonicity between the computation delay $t^{\text{comp}}_{j,i}$ and the task splitting ratio $\lambda_{j,i}$. On the one hand, we have $t^{\text{comp,e}}_{j,i} = \dfrac{\lambda_{j,i}L_{j,i}C_{j,i}}{f^{\text{e}}_{j,i}}$, which increases with $\lambda_{j,i}$. Therefore, when $\lambda_{j,i} \in [0,1]$, we can derive that $t^{\text{comp,e}}_{j,i} \in \left[0, \dfrac{L_{j,i}C_{j,i}}{f^{\text{e}}_{j,i}}\right]$. On the other hand, we have $t^{\text{tran,e}}_{j,i} + t^{\text{comp,c}}_{j,i} = (1-\lambda_{j,i})\left(\dfrac{1}{W_j} + \dfrac{C_{j,i}}{f^{\text{c}}_{j,i}}\right)L_{j,i}$, which decreases with $\lambda_{j,i}$. Thus, when $\lambda_{j,i} \in [0,1]$, we can obtain that $t^{\text{tran,e}}_{j,i} + t^{\text{comp,c}}_{j,i} \in \left[0, \left(\dfrac{1}{W_j} + \dfrac{C_{j,i}}{f^{\text{c}}_{j,i}}\right)L_{j,i}\right]$. Recall that $t^{\text{comp}}_{j,i} = \max\left\{t^{\text{comp,e}}_{j,i}, t^{\text{tran,e}}_{j,i} + t^{\text{comp,c}}_{j,i}\right\}$, therefore it first decreases and then increases with $\lambda_{j,i}$. As a result, the minimum value of $t^{\text{comp}}_{j,i}$ is achieved when $t^{\text{comp,e}}_{j,i} = t^{\text{tran,e}}_{j,i} + t^{\text{comp,c}}_{j,i}$, which results in the optimal task splitting ratio $\lambda^*_{j,i} = \dfrac{f^{\text{e}}_{j,i}\left(f^{\text{c}}_{j,i} + C_{j,i}W_j\right)}{f^{\text{e}}_{j,i}\left(f^{\text{c}}_{j,i} + C_{j,i}W_j\right) + f^{\text{c}}_{j,i}C_{j,i}W_j} = \dfrac{\eta_{j,i} + \gamma_{j,i}}{\eta_{j,i} + \gamma_{j,i} + \eta_{j,i}\gamma_{j,i}}$, where $\eta_{j,i} = \dfrac{C_{j,i}W_{j,i}}{f^{\text{e}}_{j,i}}$ and $\gamma_{j,i} = \dfrac{f^{\text{c}}_{j,i}}{f^{\text{e}}_{j,i}}$. This ends the proof.

APPENDIX D
PROOF OF LEMMA 2

To prove Lemma 2, we shall verify that the objective function and the constraints of $\mathcal{P}_4$ are all convex. It can be seen that the constraints (7c) and (7d) are affine, which reflects their convexity. In the following, we will prove that (17) is also a convex function.

The Hessian of $\widehat{t^{\text{comp}}_{j,i}}$ can be characterized as

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2\widehat{t^{\text{comp}}_{j,i}}}{\partial(f^{\text{e}}_{j,i})^2} & \dfrac{\partial^2\widehat{t^{\text{comp}}_{j,i}}}{\partial f^{\text{e}}_{j,i}\partial f^{\text{c}}_{j,i}} \\ \dfrac{\partial^2\widehat{t^{\text{comp}}_{j,i}}}{\partial f^{\text{c}}_{j,i}\partial f^{\text{e}}_{j,i}} & \dfrac{\partial^2\widehat{t^{\text{comp}}_{j,i}}}{\partial(f^{\text{c}}_{j,i})^2} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}. \tag{27}$$

With simple mathematical calculation, we can obtain all the leading principal minors of $\mathbf{H}$, as

$$\Delta_1 = H_{11} = \frac{2C_{j,i}\left(C_{j,i}W_j + f^{\text{c}}_{j,i}\right)^3 L_{j,i}}{\left(f^{\text{c}}_{j,i}f^{\text{e}}_{j,i} + C_{j,i}W_j\left(f^{\text{c}}_{j,i} + f^{\text{e}}_{j,i}\right)\right)^3} > 0, \tag{28}$$

$$\Delta_2 = H_{11}H_{22} - H_{12}H_{21} \tag{29}$$

$$= \frac{4C^4_{j,i}W^2_j\left(C_{j,i}W_j + f^{\text{c}}_{j,i}\right)^2 L^2_{j,i}}{\left(f^{\text{c}}_{j,i}f^{\text{e}}_{j,i} + C_{j,i}W_j\left(f^{\text{c}}_{j,i} + f^{\text{e}}_{j,i}\right)\right)^5} > 0. \tag{30}$$

According to linear algebra theory, $\mathbf{H}$ is positive-definite based on the fact that all its leading principal minors are positive. As a result, $\widehat{t^{\text{comp}}_{j,i}}$ is convex on both $f^{\text{e}}_{j,i}$ and $f^{\text{c}}_{j,i}$. Furthermore, the objective function of $\mathcal{P}_4$, i.e., $\sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i}\widehat{t^{\text{comp}}_{j,i}}$ is the weighted summation of a series of convex functions, therefore it is also convex [44]. To sum up, $\mathcal{P}_4$ is a classical convex optimization problem.

APPENDIX E
PROOF OF THEOREM 3

The partial Lagrange function for $\mathcal{P}_4$ can be defined as

$$\begin{aligned} L = &\sum_{j=1}^{J}\sum_{i=1}^{I_j}\beta_{j,i}\widehat{t_{j,i}^{\text{comp}}} + \sum_{j=1}^{J}\mu_j\left(\sum_{i=1}^{I_j}f_{j,i}^{\text{e}} - F_j^{\text{e}}\right) \\ &+ \theta\left(\sum_{j=1}^{J}\sum_{i=1}^{I_j}f_{j,i}^{\text{c}} - F^{\text{c}}\right), \end{aligned} \tag{31}$$

where $\theta \geq 0$ and $\mu_j \geq 0$, $\forall j \in \mathcal{J}$ are the Lagrange multipliers associated with constraints (7c) and (7d), respectively. Let $\left\{f_{j,i}^{\text{e}*}, f_{j,i}^{\text{c}*}\right\}$ be the optimal solution to $\mathcal{P}_4$. Due to the fact that the edge computation resource $f_{j,i}^{\text{e}}$ and the cloud computation resource $f_{j,i}^{\text{c}}$ are both non-negative, we can derive the following necessary and sufficient conditions, as

$$\begin{aligned} \frac{\partial L}{\partial f_{j,i}^{\text{e}*}} = &-\beta_{j,i}L_{j,i}C_{j,i}\left(\frac{f_{j,i}^{\text{c}*} + C_{j,i}W_j}{f_{j,i}^{\text{e}*}f_{j,i}^{\text{c}*} + C_{j,i}W_j\left(f_{j,i}^{\text{e}*} + f_{j,i}^{\text{c}*}\right)}\right)^2 \\ &+ \mu_j^*\begin{cases} \geq 0, & f_{j,i}^{\text{e}*} = 0, \\ = 0, & f_{j,i}^{\text{e}*} > 0, \end{cases} \end{aligned} \tag{32}$$

$$\begin{aligned} \frac{\partial L}{\partial f_{j,i}^{\text{c}*}} = &-\frac{\beta_{j,i}L_{j,i}C_{j,i}^3W_j^2}{\left(f_{j,i}^{\text{e}*}f_{j,i}^{\text{c}*} + C_{j,i}W_j\left(f_{j,i}^{\text{e}*} + f_{j,i}^{\text{c}*}\right)\right)^2} \\ &+ \theta^*\begin{cases} \geq 0, & f_{j,i}^{\text{c}*} = 0, \\ = 0, & f_{j,i}^{\text{c}*} > 0, \end{cases} \end{aligned} \tag{33}$$

$$\mu_j^*\left(\sum_{i=1}^{I_j}f_{j,i}^{\text{e}*} - F_j^{\text{e}}\right) = 0,\ \sum_{i=1}^{I_j}f_{j,i}^{\text{e}*} \leq F_j^{\text{e}},\ \mu_j^* \geq 0,\ \forall j \in \mathcal{J}, \tag{34}$$

$$\theta^*\left(\sum_{j=1}^{J}\sum_{i=1}^{I_j}f_{j,i}^{\text{c}*} - F^{\text{c}}\right) = 0,\ \sum_{j=1}^{J}\sum_{i=1}^{I_j}f_{j,i}^{\text{c}*} \leq F^{\text{c}},\ \theta^* \geq 0. \tag{35}$$

Based on the above conditions, we can obtain the optimal computation resource allocation policy, as

$$\begin{cases} f_{j,i}^{\text{e}*} = \left[\sqrt{\dfrac{\beta_{j,i}L_{j,i}C_{j,i}}{\mu_j^*}} - \left(1 - \sqrt{\dfrac{\theta^*}{\mu_j^*}}\right)C_{j,i}W_j\right]^+, \\ f_{j,i}^{\text{c}*} = \left(\sqrt{\dfrac{\mu_j^*}{\theta^*}} - 1\right)^+ C_{j,i}W_j, \end{cases} \tag{36}$$

This ends the proof.

APPENDIX F
PROOF OF COROLLARY 1

To prove Corollary 1, we shall first define some auxiliary variables, as

- Define a set $\mathcal{K} \subset \mathcal{J}$, which contains all edge nodes whose tasks are processed at the MEC servers only, without offloading to the cloud server.
- Define J subsets $\mathcal{D}_j, \forall j \in \mathcal{J}$. Each subset $\mathcal{D}_j$ contains the mobile devices connected with the j-th edge node, whose tasks are all offloaded to the cloud server, without processing at the edge node. It should be noted that for all $k \in \mathcal{K}$, the corresponding set $\mathcal{D}_k = \emptyset$.

After that, we can prove Corollary 1 as follows. By combining Theorem 3 with the above definitions, we can rewrite the optimal computation resource allocation policy as

$$f_{j,i}^{\text{e}*} = \begin{cases} \sqrt{\dfrac{\beta_{j,i}L_{j,i}C_{j,i}}{\mu_j^*}} - \left(1 - \sqrt{\dfrac{\theta^*}{\mu_j^*}}\right)C_{j,i}W_j, \\ \qquad\qquad \forall i \notin \mathcal{D}_j,\ j \in \mathcal{J}, \\ 0, \qquad\qquad \forall i \in \mathcal{D}_j,\ j \in \mathcal{J}, \end{cases} \tag{37}$$

$$f_{j,i}^{\text{c}*} = \begin{cases} \left(\sqrt{\dfrac{\mu_j^*}{\theta^*}} - 1\right)C_{j,i}W_j, & \forall j \notin \mathcal{K}, \\ 0, & \forall j \in \mathcal{K}. \end{cases} \tag{38}$$

Inserting (37) and (38) into both (34) and (35), we can derive the optimal Lagrange multipliers

$$\mu_j^* = \left(\frac{M_j + \sqrt{\theta^*}N_j}{F_j^{\text{e}} + N_j}\right)^2,\ \forall j \in \mathcal{J}, \tag{39}$$

$$\theta^* = \left(\frac{\sum_{j=1,j\notin\mathcal{K}}^{J}\dfrac{M_jS_j}{F_j^{\text{e}} + N_j}}{F^{\text{c}} + \sum_{j=1,j\notin\mathcal{K}}^{J}\dfrac{F_j^{\text{e}}S_j}{F_j^{\text{e}} + N_j}}\right)^2, \tag{40}$$

where $M_j = \sum_{i=1,i\notin\mathcal{D}_j}^{I_j}\sqrt{\beta_{j,i}L_{j,i}C_{j,i}}$, $N_j = \sum_{i=1,i\notin\mathcal{D}_j}^{I_j}C_{j,i}W_j$, and $S_j = \sum_{i=1}^{I_j}C_{j,i}W_j$, $\forall j \in \mathcal{J}$.

For each edge node $k \in \mathcal{K}$, the associated Lagrange multipliers satisfy $\mu_k^* \leq \theta^*$. Meanwhile, the Lagrange multipliers of other edge nodes satisfy $\mu_j^* > \theta^*$, where $j \notin \mathcal{K}$. Therefore, inserting (39) and (40) into $\mu_k^* \leq \theta^*$, we can obtain

$$\begin{aligned} F_k^{\text{e}} &\geq \frac{M_k\left(F^{\text{c}} + \sum_{j=1,j\notin\mathcal{K}}^{J}\dfrac{F_j^{\text{e}}S_j}{F_j^{\text{e}} + N_j}\right)}{\sum_{j=1,j\notin\mathcal{K}}^{J}\dfrac{M_jS_j}{F_j^{\text{e}} + N_j}} \\ &\geq \frac{M_k}{\sum_{j=1,j\notin\mathcal{K}}^{J}\dfrac{M_jS_j}{F_j^{\text{e}} + N_j}}F^{\text{c}} \geq \frac{M_k}{\sum_{j=1,j\notin\mathcal{K}}^{J}M_j\dfrac{S_j}{F_j^{\text{e}}}}F^{\text{c}} \\ &\geq \frac{M_k}{\max_{j=1,j\notin\mathcal{K}}^{J}\left\{\dfrac{S_j}{F_j^{\text{e}}}\right\}\sum_{j=1,j\notin\mathcal{K}}^{J}M_j}F^{\text{c}} \\ &\geq \min_{j=1}^{J}\left\{\frac{F_j^{\text{e}}}{S_j}\right\}\frac{\sum_{i=1}^{I_k}\sqrt{\beta_{k,i}L_{k,i}C_{k,i}}}{\sum_{j=1}^{J}\sum_{i=1}^{I_j}\sqrt{\beta_{j,i}L_{j,i}C_{j,i}}}F^{\text{c}}. \end{aligned} \tag{41, 42}$$

This ends the proof.

REFERENCES

[1] J. Ren, Y. He, G. Yu, and G. Y. Li, "Joint communication and computation resource allocation for cloud-edge collaborative system," in *Proc. IEEE Wireless Commun. Networking Conf. (WCNC)*, Marrakech, Morocco, Apr. 2019, pp. 1-6.

[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] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54-61, Apr. 2017.

[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] 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.

[7] 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.

[8] 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.

[9] X. Xiang, C. Lin, and X. Chen, "Energy-efficient link selection and transmission scheduling in mobile cloud computing," *IEEE Wireless Commun. Lett.*, vol. 3, no. 2, pp. 153-156, Apr. 2014.

[10] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, "A game theoretic resource allocation for overall energy minimization in mobile cloud computing system," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design*, Redondo Beach, CA, Jul.-Aug. 2012, pp. 279-284.

[11] 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.

[12] 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.

[13] 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.

[14] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2253-2266, Aug. 2015.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 998-1010, April 2018.

[20] J. Zhang et al., "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching," *IEEE Internet Things J.*, early access.

[21] L. Yang, B. Liu, J. Cao, Y. Sahni, and Z. Wang, "Joint computation partitioning and resource allocation for latency sensitive applications in mobile edge clouds," *IEEE Trans. Serv. Comput*, early access.

[22] 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.

[23] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506-5519, Aug. 2018.

[24] X. Masip-Bruin, E. Marłn-Tordera, G. Tashakor, A. Jukan, and G. J. Ren, "Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 120-128, Oct. 2016.

[25] V. B. C. Souza, W. Ramłrez, X. Masip-Bruin, E. Marłn-Tordera, G. Ren, and G. Tashakor, "Handling service allocation in combined fog-cloud scenarios," in *Proc. IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1-5.

[26] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, San Francisco, California, Apr. 2016, pp. 1-9.

[27] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: A computation offloading game," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3246-3257, Aug. 2018.

[28] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171-1181, Dec. 2016.

[29] M. H. Chen, B. Liang, and M. Dong, "Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Atlanta, Georgia, May 2017, pp. 1-9.

[30] T. Z. Oo, N. H. Tran, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Offloading in HetNet: A coordination of interference mitigation, user association, and resource allocation," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2276-2291, Aug. 2017.

[31] L. Tang, W. Wang, Y. Wang, and Q. Chen, "An energy-saving algorithm with joint user association, clustering, and on/off strategies in dense heterogeneous networks," *IEEE Access*, vol. 5, pp. 12988-13000, Jul. 2017.

[32] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.

[33] 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.

[34] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Luo, "Fog computing based face identification and resolution scheme in internet of things," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1910-1920, Aug. 2017.

[35] 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 International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1-6.

[36] X. Meng, W. Wang, and Z. Zhang, "Delay-constrained hybrid computation offloading with cloud and fog computing," *IEEE Access*, vol. 5, pp. 21355-21367, 2017.

[37] J. Ren, Y. He, G. Huang, G. Yu, Y. Cai, and Z. Zhang, "An edge-computing based architecture for mobile augmented reality," *IEEE Network*, early access.

[38] C. Liu, L. Zhang, M. Zhu, J. Wang, L. Cheng, and G. K. Chang, "A novel multi-service small-cell cloud radio access network for mobile backhaul and computing based on radio-over-fiber technologies," *J. Lightw. Technol.*, vol. 31, no. 17, pp. 2869-2875, Sep. 2013.

[39] G. Zhang, T. Q. S. Quek, M. Kountouris, A. Huang, and H. Shan, "Fundamentals of heterogeneous backhaul design-analysis and optimization," *IEEE Trans. Commun.,* vol. 64, no. 2, pp. 876-889, Feb. 2016.

[40] A. Al-Shuwaili, O. Simeone, A. Bagheri, and G. Scutari, "Joint uplink/downlink optimization for backhaul-limited mobile cloud computing with user scheduling," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 4, pp. 787-802, Dec. 2017.

[41] X. Peng, J. Zhang, S. H. Song, and K. B. Letaief, "Cache size allocation in backhaul limited wireless networks," in *Proc. IEEE International Conference on Communications (ICC)*, Kuala Lumpur, May 2016, pp. 1-6.

[42] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidisc. Optim.*, vol. 26, no. 6, pp. 369-395, Apr. 2004.

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

[44] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

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, and a lead guest editor of *IEEE Wireless Communications Magazine* special issue on LTE in Unlicensed Spectrum, and an Editor of IEEE Access. 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 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 also serves as a Symposium Co-Chair for IEEE Globecom 2019 and a Track Chair for IEEE VTC 2019'Fall.

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

Geoffrey Ye Li (S'93-M'15-SM'97-F'06) received the B.S.E. and M.S.E. degrees from the Department of Wireless Engineering, Nanjing Institute of Technology, Nanjing, China, in 1983 and 1986, respectively, and the Ph.D. degree from the Department of Electrical Engineering, Auburn University, Auburn, AL, USA, in 1994.

He was a Teaching Assistant and then a Lecturer with Southeast University, Nanjing, China, from 1986 to 1991, a Research and Teaching Assistant with Auburn University, from 1991 to 1994, and a Post-Doctoral Research Associate with the University of Maryland at College Park, College Park, MD, USA, from 1994 to 1996. He was with AT&T Labs-Research, Red Bank, NJ, USA, as a Senior and then a Principal Technical Staff Member from 1996 to 2000. Since 2000, he has been with the School of Electrical and Computer Engineering, Georgia Institute of Technology, as an Associate Professor and then a Full Professor.

His general research interests include statistical signal processing and machine learning for wireless communications. In these areas, he has published over 500 journal and conference papers in addition to over 40 granted patents. His publications have been cited over 35,000 times and he has been recognized as the *World's Most Influential Scientific Mind*, also known as a *Highly-Cited Researcher*, by Thomson Reuters almost every year. He was awarded *IEEE Fellow* for his contributions to *signal processing for wireless communications* in 2005. He won 2010 *IEEE ComSoc Stephen O. Rice Prize Paper Award*, 2013 *IEEE VTS James Evans Avant Garde Award*, 2014 *IEEE VTS Jack Neubauer Memorial Award*, 2017 *IEEE ComSoc Award for Advances in Communication*, and 2017 *IEEE SPS Donald G. Fink Overview Paper Award*. He also received 2015 Distinguished Faculty Achievement Award from the School of Electrical and Computer Engineering, Georgia Tech. He has been involved in editorial activities for over 20 technical journals for the IEEE, including founding Editor-in-Chief of IEEE 5G Tech Focus. He has organized and chaired many international conferences, including technical program vice-chair of IEEE ICC'03, technical program co-chair of IEEE SPAWC'11, general chair of IEEE GlobalSIP'14, technical program co-chair of IEEE VTC'16 (Spring), and general co-chair of IEEE VTC'19 (Fall).