D2D Communications Meet Mobile Edge Computing for Enhanced Computation Capacity in Cellular Networks

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

***Abstract*—Future 5G wireless networks aim to support high-rate data communications and high-speed mobile computing. To achieve this goal, the mobile edge computing (MEC) and device-to-device (D2D) communications have been recently developed, both of which take advantage of the proximity for better performance. In this paper, we integrate the D2D communications with MEC to further improve the computation capacity of the cellular networks where the task of each device can be offloaded to an edge node and a nearby D2D device. We aim to maximize the number of devices supported by the cellular networks with the constraints of both communication and computation resources. The optimization problem is formulated as a mixed integer non-linear problem, which is not easy to solve in general. To tackle it, we decouple it into two subproblems. The first one minimizes the required edge computation resource for a given D2D pair while the second one maximizes the number of supported devices via optimal D2D pairing. We prove that the optimal solutions to the two subproblems compose the optimal solution to the original problem. Then, the optimal algorithm to the original problem is developed by solving two subproblems and some insightful results, such as the optimal transmit power allocation and the task offloading strategy, are also highlighted. Our proposal is finally tested by extensive numerical simulation results, which demonstrate that combining D2D communications with MEC can significantly enhance the computation capacity of the system.**

***Index Terms*—Mobile edge computing, device-to-device communications, computation capacity, task offloading, resource allocation, cellular networks.**

I. Introduction

Over the past decade, the main goal of wireless communication networks is to provide high-speed data transmission and large system capacity. Aiming at this target, several appealing technologies have been recently developed and widely used in current wireless networks, such as massive multiple-input multiple-output (MIMO), device-to-device (D2D) communications, and full-duplex (FD) transmissions [1]. However, as the blooming popularity of computationally intensive mobile applications, providing high computation capacity has become another important goal for the next generation wireless networks. To meet with such a challenge, mobile edge computing (MEC) has been developed by the European Telecommunications Standard Institute (ETSI) to equip high-speed computation unit at the network edge, e.g., the cellular base station (BS) [2].

With the cloud computation capacity at the BS, mobile devices can offload their tasks to the edge cloud for processing, which is referred to as mobile edge computation offloading (MECO). The benefits of MECO are two-folds. First, by offloading computationally intensive tasks, the energy consumption of mobile devices can be significantly saved. Second, since the tasks can be computed at the adjacent BS instead of the remote cloud center, the congestion on the core network can be effectively relieved, leading to the reduction of the end-to-end latency. Several recent works have investigated various MECO problems from the perspectives of energy efficiency improvement [3]–[7], end-to-end latency minimization [8]–[11], and the tradeoff between the two objectives [12]. From the task offloading viewpoint, there are mainly two kinds of task offloading strategies. In the binary offloading [3], [4], the task can be totally offloaded to the edge cloud or totally remained for local computing. In the partial offloading [5], [6], [11], the task can be partially offloaded for edge computing and partially remained for local processing.

Although the above studies have demonstrated the effectiveness of MECO in improving the computation performance of wireless networks, the limited computation resource at the BS is not always adequate to support all mobile devices in its coverage. To deal with this issue, we are motivated to offload part of computation tasks to neighbor devices via D2D communication links. Benefiting from the proximity gain, reuse gain, and hop gain, D2D communications have been widely investigated to improve the spectrum efficiency and energy efficiency of cellular networks [13]–[15].

There have been also several works [16]–[19] that investigated the D2D offloading in the MEC system. By offloading the task to a nearby mobile device via a D2D link, the energy efficiency and delay performance of a mobile computing system can be effectively improved [16], [17]. A novel mobile task offloading framework, namely D2D fogging, has been developed in [18], where mobile devices can share the communication and computation resources for higher energy efficiency. Moreover, the non-causal CPU-state information

Manuscript received May 28, 2018; revised November 4, 2018; accepted January 28, 2019. This work was supported in part by the Natural Science Foundation of China under Grants 61671407 and 61831004; the Open Research Fund of the State Key Laboratory of Integrated Services Networks, Xidian University, under Grant ISN18-13; the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under Grant LR19F010002; and the Fundamental Research Funds for the Central Universities. The associate editor coordinating the review of this paper and approving it for publication was K. Navaie. (*Corresponding author: G. Yu.*)

Y. He 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, Xi'an 710071, China. (e-mail: {2014hyh, yuguanding}@zju.edu.cn).

J. Ren and Y. Cai are with the Zhejiang Provincial Key Laboratory of Information Processing, Communication and Networking, Zhejiang University, Hangzhou 310027, China. (e-mail: {renjinke, ylcai}@zju.edu.cn)

has been exploited to develop the optimal offloading strategy between a peer-to-peer pair [19].

The aforementioned studies are mainly from the perspective of energy efficiency maximization or end-to-end delay minimization. However, they did not focus on maximizing the computation capacity of the whole system by jointly performing D2D offloading and MECO. Inspired by this, we integrate the D2D communication into the MEC system and propose the *D2D-MEC* technique to improve the computation capacity of the system in this paper. Since the wireless communication system is designed to provide high-quality service for more devices, we adopt the number of devices that the system can support under the communication and computation resource limitations as the evaluation criterion, which can reflect the computation capacity of the system. We also study partial offloading model in this paper, where the task of each device can be partitioned into three parts with one part for local computing and rest parts for edge offloading and D2D offloading, respectively. By leveraging the advantages of both MEC and D2D communications, the computation resources in both edge cloud and D2D devices can be fully utilized and the system computation capacity can be effectively improved. The main contributions of this work are summarized as follows.

- We propose a novel D2D-MEC technique that collaborates D2D communications and MEC to improve the computation capacity of the whole system, i.e., the number of devices that the system can support. In particular, we formulate a mixed integer non-linear problem with the constraints on both communication and computation resources. By analyzing the intrinsic structure of this problem, we decompose it into two subproblems, whose optimal solutions also compose the optimal solution to the original problem.
- The first subproblem aims to minimize the required edge computation resource for a given D2D pair. To solve it, we derive closed-form expressions for the optimal transmit power allocation and the optimal task offloading strategy, respectively. Based on this, the computation gain achieved by D2D offloading is revealed and some insightful results are highlighted as well.
- The second subproblem maximizes the number of devices that the system can support based on the solutions to the first subproblem. A polynomial-complexity algorithm is developed to solve it and an upper bound and a lower bound for the optimal value are derived, respectively. By analyzing some special cases, our proposal can achieve the D2D gain and the MEC gain as compared to existing schemes.

The rest of this paper is organized as follows. In Section II, we introduce the D2D-MEC system and formulate the optimization problem to maximize the system computation capacity. In Section III, we decompose the original problem into two subproblems. The optimal solutions to two subproblems are developed in Sections IV and V, respectively. Simulation results are presented in Section VI and the whole paper is concluded in Section VII.

II. System Model and Problem Formulation

In this section, we first introduce the D2D-MEC system, and then analyze the delay of the proposed computing procedure. Afterwards, we formulate the optimization problem to maximize the system computation capacity.

A. D2D-MEC System

As depicted in Fig. 1, we consider a D2D-MEC system which consists of one BS equipped with an edge cloud and N mobile devices, denoted by the set $\mathcal{N} \triangleq \{1, 2, \cdots, N\}$. Moreover, we assume that each device has been already associated with a BS by some user association strategies in [20]. Each device can not only establish a cellular link with the BS, but also connect one and only one nearby device through a D2D link. Each device has a computation task that needs to be completed under the given delay and transmit power constraints. By following the model in [6], we utilize a tuple $\{V_n, C_n\}$ to characterize the task of device n ($n \in \mathcal{N}$), where V_n (in bits) is the size of the task data and C_n is the number of CPU cycles required for computing 1-bit data. Let T denote the maximum delay tolerance of all tasks. We also assume that the size of computation results is usually small enough so that the delay for downloading them can be neglected.

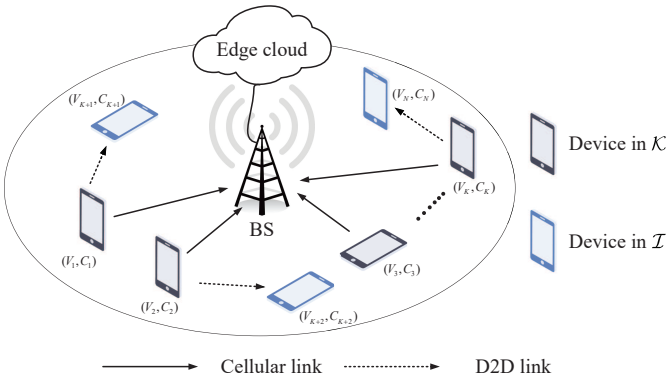


Fig. 1. The D2D-MEC system.

Let f_n (in CPU cycle/s) denote the computation resource of device n and F^{e} denote the overall computation resource at the edge cloud, which can be allocated to all devices for parallel edge computing. There are three computing procedures, i.e., local computing, D2D offloading, and edge offloading. As shown in Fig. 2, we adopt a partial offloading model, where each device can split its task into three parts with one part remained for local computing while the other two parts offloaded to the edge cloud and a neighbor device, respectively.

To facilitate our analysis, we make the following reasonable assumptions throughout this paper.

- We assume that the BS can acquire the channel state information (CSI) and the sizes of tasks of all devices via feedback. Thus, our algorithm works in a centralized manner, which provides an important reference for other practical and distributive schemes. With this assumption, some insightful results can be obtained in this work.

Nevertheless, the D2D-MEC system with imperfect CSI deserves further investigation.

- Each device will offload part of its task only if it cannot be completed on time by local computing.
- To investigate the performance limit of computation capacity, we assume that devices are willing to assist other devices after completing their own tasks on time.

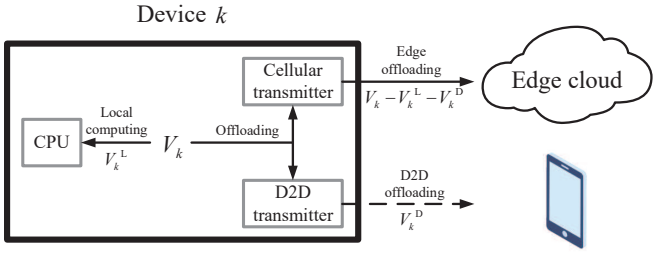


Fig. 2. Computing procedures.

Based on the above analysis, we can classify all devices into two sets according to their workload and computation resources.

- The set $\mathcal{K} \triangleq \{1, 2, \cdots, K\}$, which includes those devices that cannot complete their tasks on time by local computing, i.e., $\frac{V_k C_k}{f_k} > T, \forall k \in \mathcal{K}$.
- The set $\mathcal{I} \triangleq \{K+1, K+2, \cdots, N\}$, which includes those devices that can complete their tasks on time by local computing, i.e., $\frac{V_i C_i}{f_i} \leq T, \forall i \in \mathcal{I}$. Let I denote the size of $\mathcal{I}$, which satisfies $I + K = N$.

As mentioned earlier, each task in the system can be split into three parts. After classifying, only devices in $\mathcal{K}$ need to split their tasks. As in Fig. 2, we denote V_k^{L}, V_k^{D} as the computation workload of device k ($k \in \mathcal{K}$) for local computing and D2D offloading, respectively. Therefore, the computation workload for edge computing is $\left(V_k - V_k^{\mathrm{L}} - V_k^{\mathrm{D}}\right)$. Moreover, to meet the delay constraint for local computing, we have

$$D_k^{\mathrm{L}} = \frac{V_k^{\mathrm{L}} C_k}{f_k} \leq T, \ \forall k \in \mathcal{K}. \tag{1}$$

From (1), to reduce the required computation resource of the edge cloud, we can set $V_k^{\mathrm{L}} = \frac{T f_k}{C_k}, \forall k \in \mathcal{K}$. On the other hand, device i ($i \in \mathcal{I}$) computes its whole task locally and the local computing delay can be expressed as $D_i^{\mathrm{L}} = \frac{V_i C_i}{f_i}$, satisfying $D_i^{\mathrm{L}} \leq T$, $\forall i \in \mathcal{I}$.

Since only the device in $\mathcal{I}$ can assist the device in $\mathcal{K}$, the D2D link can be established between a device in $\mathcal{I}$ and a device in $\mathcal{K}$. First, we define $\mathcal{U}_k$ as the set of the device in $\mathcal{I}$, which can establish a D2D link with the device k in $\mathcal{K}$. Correspondingly, let $\mathcal{U}_i$ denote the set of the device in $\mathcal{K}$, which can establish a D2D link with device i in $\mathcal{I}$. Note that $\mathcal{U}_i$ and $\mathcal{U}_k$ can be determined by the location of devices, e.g., we can simply assume that device i and device k can establish a D2D communication link if their distance is within a given threshold, i.e., R. In addition, the value of R is determined by specific configuration of each device. It is intuitive that a larger R would lead to a better system computation capacity since more devices in $\mathcal{I}$ can be potentially paired by the devices in $\mathcal{K}$. Then, we define a binary decision variable $u_{k,i}$ to characterize the connectivity between the devices in set $\mathcal{K}$ and set $\mathcal{I}$. That is, $u_{k,i}$ equals one when device k establishes a D2D link with device i and offloads its task via this D2D link, and zero otherwise.

As for edge offloading, denote f_k^{e} as the computation resource that the edge cloud allocates to device k. Then, f_k^{e} also represents the indicator for edge offloading, which is positive if device k offloads its task to the edge cloud and zero otherwise.

Accordingly, we can utilize π_k to describe whether device k can complete its task on time, which means device k fails when $\pi_k = 0$, and succeeds when $\pi_k = 1$. However, there is a special case that has not been considered, where device k is allocated enough edge computation resource and completes its task without D2D offloading. Hence, we introduce $u_{k,0}$ to characterize this special case where $u_{k,0} = 1$ means device k only utilizes edge offloading and local computing to complete its task and zero otherwise. Therefore, we can calculate π_k as $\pi_k = \sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i}$.

From the above definitions, we can summarize the different meanings with different $u_{k,i}$ and f_k^{e}. That is, there is at most one element in the kth row of $\{u_{k,i}\}$ equals one while the others are all zero. If $u_{k,0} = 1$, f_k^{e} must be bigger than zero and the device k will only offload its task to the edge cloud without D2D offloading. If $u_{k,i} = 1, i > 0$, the device k will offload its task to the device i via a D2D link. In this case, whether the edge offloading is required depends on whether $f_k^{\mathrm{e}} > 0$ or $f_k^{\mathrm{e}} = 0$. If $u_{k,i} = 0, \forall i \in \mathcal{U}_k \cup \{0\}$, device k fails to complete its task on time.

B. Wireless Channel Model

We adopt orthogonal frequency-division multiple access (OFDMA) method for channel access. Each device in $\mathcal{K}$ is allocated with one sub-channel for the cellular link, and each D2D link is also allocated with one sub-channel, respectively. It is reasonable to assume that each device can establish one cellular link and one D2D link simultaneously according to [21]. Denote B and N_0 as the bandwidth and the noise power for each sub-channel, respectively. Let h_k^{C} denote the channel power gain of cellular link for device k and p_k^{C} denote the transmit power for edge offloading. Then the achievable data rate of each cellular link is given by

$$r_k^{\mathrm{C}}(p_k^{\mathrm{C}}) = B \log_2 \left(1 + \frac{p_k^{\mathrm{C}} \left|h_k^{\mathrm{C}}\right|^2}{N_0}\right), \forall k \in \mathcal{K}. \tag{2}$$

Similarly, let $h_{k,i}$ denote the channel power gain between device k and device i and p_k^{D} denote the transmit power for D2D offloading. Then, the achievable data rate of the D2D pair (k, i) is given by

$$r_{k,i}^{\mathrm{D}}(p_k^{\mathrm{D}}) = B \log_2 \left(1 + \frac{p_k^{\mathrm{D}} \left|h_{k,i}\right|^2}{N_0}\right), \forall i \in \mathcal{U}_k, \forall k \in \mathcal{K}. \tag{3}$$

Additionally, the total transmit power of device k is limited by $p_k^{\max}$, that is $p_k^{\mathrm{D}} + p_k^{\mathrm{C}} \le p_k^{\max}, \forall k \in \mathcal{K}$.

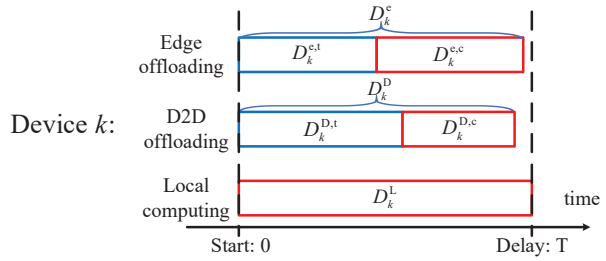
C. Delay Analysis

As mentioned earlier, the device k ($k \in \mathcal{K}$) would offload V_k^{D}-bit data to device i ($i \in \mathcal{I}$) via the D2D link and $\left(V_k - V_k^{\mathrm{L}} - V_k^{\mathrm{D}}\right)$-bit data to the edge cloud via the cellular link. Therefore, as depicted in Fig. 3(a), there are four kinds of delays, as

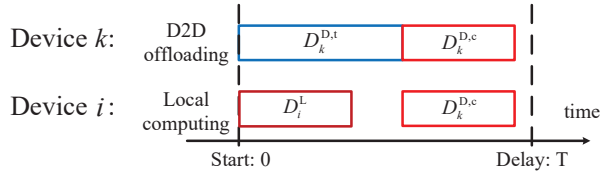
- The delay for D2D transmitting, $D_k^{\mathrm{D,t}} = \frac{V_k^{\mathrm{D}}}{\sum_{i\in\mathcal{U}_k} u_{k,i} r_{k,i}^{\mathrm{D}}(p_k^{\mathrm{D}})}$.
- The delay for D2D computing, $D_k^{\mathrm{D,c}} = \frac{V_k^{\mathrm{D}} C_k}{\sum_{i\in\mathcal{U}_k} u_{k,i} f_i}$.
- The delay for edge transmitting, $D_k^{\mathrm{e,t}} = \frac{V_k - V_k^{\mathrm{L}} - V_k^{\mathrm{D}}}{r_k^{\mathrm{C}}(p_k^{\mathrm{C}})}$.
- The delay for edge computing, $D_k^{\mathrm{e,c}} = \frac{\left(V_k - V_k^{\mathrm{L}} - V_k^{\mathrm{D}}\right) C_k}{f_k^{\mathrm{e}}}$.

Therefore, the overall delay for edge offloading is given by

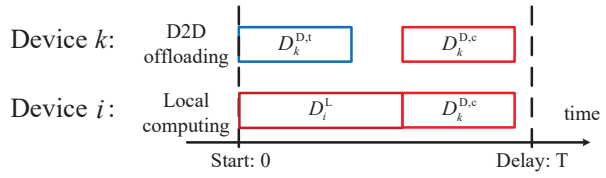
$$D_k^{\mathrm{e}} = D_k^{\mathrm{e,t}} + D_k^{\mathrm{e,c}}, \forall k \in \mathcal{K}. \tag{4}$$



(a) Delays for local computing, D2D offloading, and edge offloading of device k.



(b) Delays for D2D offloading of device k. Case 1: the delay for D2D transmission is dominant.



(c) Delays for D2D offloading of device k. Case 2: the delay for local computing is dominant.

Fig. 3. Delay analysis of device k.

Regarding the overall delay for D2D offloading, the delay for the local computing of device i ($i \in \mathcal{I}$) should be counted since this device can help others only when its own task is completed. We further assume the data from device k cannot be processed by device i until the D2D transmission is completed. Therefore, as shown in Fig. 3(b) and 3(c), the beginning of D2D computing is determined by the larger one between the delay for local computing of device i and the delay for D2D transmission of device k. Therefore, the overall delay for D2D offloading of device k can be expressed as

$$D_k^{\mathrm{D}} = \max\left(D_k^{\mathrm{D,t}}, \sum_{i\in\mathcal{U}_k} u_{k,i} D_i^{\mathrm{L}}\right) + D_k^{\mathrm{D,c}}, \forall k \in \mathcal{K}. \tag{5}$$

D. Problem Formulation

In this paper, we aim at maximizing the computation capacity of the considered D2D-MEC system, where each device in $\mathcal{K}$ can split its task into three parts for local computing, D2D offloading and edge offloading, simultaneously. Specifically, the computation capacity is defined as the number of supported devices whose tasks can be completed within the required time limit, i.e., T. Note that the devices in $\mathcal{I}$ can complete their tasks by local computing only. Therefore, we pay our special attention to the devices in $\mathcal{K}$. Based on the above analysis, the objective function can be modelled as $\sum_{k=1}^{K} \pi_k = \sum_{k=1}^{K} \sum_{i\in\mathcal{U}_k\cup\{0\}} u_{k,i}$ and the mathematical problem to maximize the computation capacity of the D2D-MEC system can be formulated as

$$\mathcal{P}1: \max_{\substack{\{u_{k,i},\pi_k,\\ V_k^{\mathrm{D}},p_k^{\mathrm{C}},\\ p_k^{\mathrm{D}},f_k^{\mathrm{e}}\}}} \sum_{k=1}^{K} \pi_k, \tag{6a}$$

$$\text{s.t.} \quad \pi_k \max\left(D_k^{\mathrm{e}}, D_k^{\mathrm{D}}\right) \le T, \ \forall k \in \mathcal{K}, \tag{6b}$$

$$p_k^{\mathrm{C}} + p_k^{\mathrm{D}} \le p_k^{\max}, \ \forall k \in \mathcal{K}, \tag{6c}$$

$$0 \le V_k - V_k^{\mathrm{L}} - V_k^{\mathrm{D}}, \ \forall k \in \mathcal{K}, \tag{6d}$$

$$\sum_{k=1}^{K} \pi_k f_k^{\mathrm{e}} \le F^{\mathrm{e}}, \tag{6e}$$

$$\sum_{i\in\mathcal{U}_k} u_{k,i} \le 1, \ \forall k \in \mathcal{K}, \tag{6f}$$

$$\sum_{k\in\mathcal{U}_i} u_{k,i} \le 1, \ \forall i \in \mathcal{I}, \tag{6g}$$

$$p_k^{\mathrm{C}}, p_k^{\mathrm{D}}, f_k^{\mathrm{e}}, V_k^{\mathrm{D}} \ge 0, \ \forall k \in \mathcal{K}, \tag{6h}$$

$$u_{k,i}, \pi_k \in \{0,1\}, \ \forall i \in \mathcal{U}_k \cup \{0\}, \ \forall k \in \mathcal{K}. \tag{6i}$$

In the above, constraint (6b) guarantees the time limitation for edge offloading and D2D offloading, respectively. Constraint (6c) bounds the maximum transmit power of each device. Constraint (6d) guarantees that the size of task offloaded to the edge cloud is no less than zero. Constraint (6e) is the total computation resource limitation of the edge cloud. Constraints (6f) and (6g) guarantee that each device can only establish one D2D link. The optimization variables in $\mathcal{P}1$ contain the D2D device pairing indicator ($u_{k,i}$ and π_k), the computation resource allocation at the edge cloud (f_k^{e}), the power allocation of each device (p_k^{D} and p_k^{C}), and the task split strategy (V_k^{D}). It can be observed that $\mathcal{P}1$ is a mixed integer non-linear problem, which is hard to solve in general. In the next section, we will decompose it into two subproblems.

III. Problem Decomposition

The main challenge in solving $\mathcal{P}1$ is that both combinational variables $\{u_{k,i}, \pi_k\}$ and continuous variables $\left\{f_k^{\mathrm{e}}, p_k^{\mathrm{D}}, p_k^{\mathrm{C}}, V_k^{\mathrm{D}}\right\}$ are involved. However, by analyzing the problem, we can successfully decouple it into two subproblems and then solve them individually.

Since our goal is to maximize the number of supported devices in the system, and the main restriction of $\mathcal{P}1$ is the limited computation resource of the edge cloud, we can first minimize the required edge computation resource for each given D2D pair (k,i) by jointly optimizing the task splitting variable V_k^{D} and the power allocation $\left\{p_k^{\mathrm{D}}, p_k^{\mathrm{C}}\right\}$. Let $f_{k,i}^{\mathrm{e}}$ denote the required edge computation resource that guarantees device k to complete its task on time when it connects device i via a D2D link. The mathematical formulation for minimizing the required edge computation resource can be expressed as

$$\mathcal{P}2: \quad \min_{\left\{\substack{f_{k,i}^{\mathrm{e}}, p_k^{\mathrm{D}}, \\ p_k^{\mathrm{C}}, V_k^{\mathrm{D}}}\right\}} \quad f_{k,i}^{\mathrm{e}}, \tag{7a}$$

$$\text{s.t.} \quad \left(V_k - V_k^{\mathrm{D}} - V_k^{\mathrm{L}}\right) - T\left(\frac{1}{r_k^{\mathrm{C}}(p_k^{\mathrm{C}})} + \frac{C_k}{f_{k,i}^{\mathrm{e}}}\right)^{-1} \leq 0, \tag{7b}$$

$$V_k^{\mathrm{D}} - T\left(\frac{1}{r_{k,i}^{\mathrm{D}}(p_k^{\mathrm{D}})} + \frac{C_k}{f_i}\right)^{-1} \leq 0, \tag{7c}$$

$$\frac{V_i C_i + V_k^{\mathrm{D}} C_k}{f_i} - T \leq 0, \tag{7d}$$

$$p_k^{\mathrm{D}} + p_k^{\mathrm{C}} \leq p_k^{\max}, \tag{7e}$$

$$0 \leq V_k - V_k^{\mathrm{L}} - V_k^{\mathrm{D}}, \tag{7f}$$

$$p_k^{\mathrm{D}}, p_k^{\mathrm{C}}, f_{k,i}^{\mathrm{e}}, V_k^{\mathrm{D}} \geq 0. \tag{7g}$$

In the above, we have transformed constraint (6b) into (7b) - (7d) to make $\mathcal{P}2$ a convex optimization problem. Note that this subproblem only contains continuous variables and the optimal solution will be developed in Section IV.

After obtaining the minimum required edge computation resource, denoted by $f_{k,i}^{\mathrm{e},*}$, by solving $\mathcal{P}2$ for each potential D2D pair, we can now maximize the number of supported devices by optimizing the D2D pairing. Therefore, the original problem, i.e., $\mathcal{P}1$, can be reformulated as

$$\mathcal{P}3: \quad \max_{\{u_{k,i}, \pi_k\}} \quad \sum_{k=1}^{K} \pi_k, \tag{8a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i} f_{k,i}^{\mathrm{e},*} \leq F^{\mathrm{e}}, \tag{8b}$$

$$\text{(6f) and (6g)},$$

$$u_{k,i}, \pi_k \in \{0,1\}, \ \forall i \in \mathcal{U}_k \cup \{0\}, \ \forall k \in \mathcal{K}, \tag{8c}$$

where the constraint (6e) is rewritten into (8b) since $f_k^{\mathrm{e}} = \sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i} f_{k,i}^{\mathrm{e},*}$. It can be easily observed that $\mathcal{P}3$ is an integer linear programming problem, which will be solved in Section V.

By analyzing the intrinsic relations between the two subproblems and the original problem, we have the following theorem.

Theorem 1: The optimal solutions to $\mathcal{P}2$ and $\mathcal{P}3$ compose the optimal solution to $\mathcal{P}1$.

Proof: Please see Appendix A. ∎

IV. Computation Resource Minimization

In this section, we will solve $\mathcal{P}2$, i.e., minimizing the required edge computation resource for a given D2D pair (k, i), and derive the optimal power allocation and data offloading strategy. Some important insights will be also highlighted. Finally, a special case where each device processes its task by local computing and edge offloading will be investigated.

A. The Optimal Solution

Lemma 1: $\mathcal{P}2$ is a convex optimization problem.

Proof: Please see Appendix B. ∎

According to Lemma 1, we can solve $\mathcal{P}2$ by leveraging the Lagrangian method. The partial Lagrangian function can be defined as

$$\begin{aligned} L = {} & f_{k,i}^{\mathrm{e}} + \mu\left(V_k^{\mathrm{D}} - T\left(\frac{1}{r_{k,i}^{\mathrm{D}}(p_k^{\mathrm{D}})} + \frac{C_k}{f_i}\right)^{-1}\right) \\ & + \lambda\left(\left(V_k - V_k^{\mathrm{D}} - V_k^{\mathrm{L}}\right) - T\left(\frac{1}{r_k^{\mathrm{C}}(p_k^{\mathrm{C}})} + \frac{C_k}{f_{k,i}^{\mathrm{e}}}\right)^{-1}\right) \\ & + \omega\left(p_k^{\mathrm{D}} + p_k^{\mathrm{C}} - p_k^{\max}\right), \end{aligned} \tag{9}$$

where λ, μ, and ω are the Lagrange multipliers associated with the constraints (7b), (7c), and (7e), respectively. Let $\left\{f_{k,i}^{\mathrm{e},*}, p_k^{\mathrm{D},*}, p_k^{\mathrm{B},*}, V_k^{\mathrm{D},*}\right\}$ denote the optimal solution to $\mathcal{P}2$. Then, based on the Karush-Kuhn-Tucker (KKT) conditions, we can find the optimal solution later. Note that constraints (7d), (7f), and (7g) will be included in KKT conditions in Appendix C.

To gain some insights, we shall first define some auxiliary functions and values as follows.

- The maximal D2D transmit power,

$$p_{k,i}^{\mathrm{D},\max} \triangleq \min\left(\rho_{k,i}\left(\frac{T f_i^2}{V_i C_i C_k} - \frac{f_i}{C_k}\right), p_k^{\max}\right), \tag{10}$$

  where $\rho_{k,i}(x)$ is the inverse function of $r_{k,i}^{\mathrm{D}}\left(p_k^{\mathrm{D}}\right)$. It gives the limitation on each D2D link's transmit power and is jointly determined by constraints (7c), (7d), and (7e).
- The D2D data offloading size function,

$$V_{k,i}^{\mathrm{D}}\left(p^{\mathrm{D}}\right) \triangleq T\left(\frac{1}{r_{k,i}^{\mathrm{D}}\left(p^{\mathrm{D}}\right)} + \frac{C_k}{f_i}\right)^{-1}, \tag{11}$$

  which represents the data size via D2D offloading under the delay constraint when the transmit power of the D2D link is p^{D}.

- The transmit power allocation function,

$$\varphi_{k,i}\left(p^{\mathrm{D}}\right) \triangleq \frac{\left(V_k - V_k^{\mathrm{L}} - V_{k,i}^{\mathrm{D}}\left(p^{\mathrm{D}}\right)\right)^2}{\left(r_k^{\mathrm{C}}\left(p_k^{\max} - p^{\mathrm{D}}\right)\right)^2} \frac{\mathrm{d}r_k^{\mathrm{C}}\left(p_k^{\mathrm{C}}\right)}{\mathrm{d}p_k^{\mathrm{C}}}\Big|_{p_k^{\mathrm{C}}=p_k^{\max}-p^{\mathrm{D}}} - \frac{\left(V_{k,i}^{\mathrm{D}}\left(p^{\mathrm{D}}\right)\right)^2}{\left(r_{k,i}^{\mathrm{D}}\left(p^{\mathrm{D}}\right)\right)^2} \frac{\mathrm{d}r_{k,i}^{\mathrm{D}}\left(p_k^{\mathrm{D}}\right)}{\mathrm{d}p_k^{\mathrm{D}}}\Big|_{p_k^{\mathrm{D}}=p^{\mathrm{D}}}, \quad (12)$$

  which represents the difference of the power gain between D2D offloading and edge offloading and it is a function of the D2D transmit power, i.e., p^{D}. Besides, we define $\Phi_{k,i}(x)$ as the inverse function of $\varphi_{k,i}\left(p^{\mathrm{D}}\right)$.

- The total offloading data size,

$$V_{k,i}^{\max} \triangleq \max_{p^{\mathrm{D}}\in\left[0,p_{k,i}^{\mathrm{D,max}}\right]} \left(\left(\frac{1}{r_{k,i}^{\mathrm{D}}\left(p^{\mathrm{D}}\right)} + \frac{C_k}{f_i}\right)^{-1} + r_k^{\mathrm{C}}\left(p_k^{\max} - p^{\mathrm{D}}\right)\right) T, \quad (13)$$

  which represents the overall data size via D2D offloading and edge offloading when the edge computation resource allocated to device k is infinite.

With the above definitions, we can derive the optimal solution to $\mathcal{P}2$ with the aid of KKT conditions and simple mathematical analysis, as presented in the following theorem.

Theorem 2: The optimal solution to $\mathcal{P}2$ can be classified into three cases, which are summarized as follows.

Case 1: If $V_k \le V_{k,i}^{\mathrm{D}}\left(p_{k,i}^{\mathrm{D,max}}\right) + V_k^{\mathrm{L}}$, the optimal solution is given by

$$\begin{cases} \left(p_k^{\mathrm{D},*}, V_k^{\mathrm{D},*}\right) = \left(p_k^{\max}, V_k - V_k^{\mathrm{L}}\right), \\ p_k^{\mathrm{C},*} = 0, \\ f_{k,i}^{\mathrm{e},*} = 0. \end{cases} \quad (14)$$

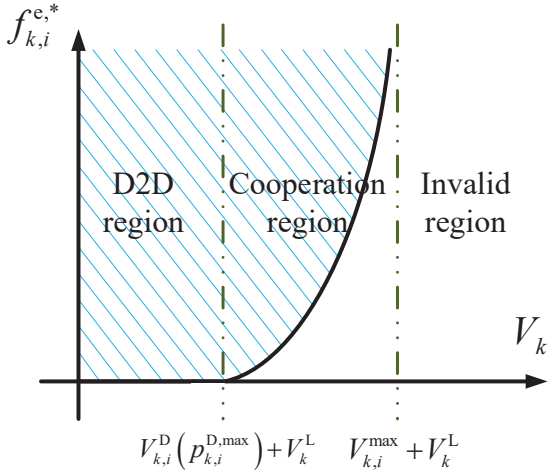
Case 2: If $V_{k,i}^{\mathrm{D}}\left(p_{k,i}^{\mathrm{D,max}}\right) + V_k^{\mathrm{L}} < V_k < V_{k,i}^{\max} + V_k^{\mathrm{L}}$, the optimal solution is given by

$$\begin{cases} \left(p_k^{\mathrm{D},*}, V_k^{\mathrm{D},*}\right) = \left(\min\left(\Phi_{k,i}\left(0\right), p_{k,i}^{\mathrm{D,max}}\right), V_{k,i}^{\mathrm{D}}\left(p_k^{\mathrm{D},*}\right)\right), \\ p_k^{\mathrm{C},*} = p_k^{\max} - p_k^{\mathrm{D},*}, \\ f_{k,i}^{\mathrm{e},*} = C_k\left(\frac{T}{V_k - V_k^{\mathrm{L}} - V_k^{\mathrm{D},*}} - \frac{1}{r_k^{\mathrm{C}}\left(p_k^{\mathrm{C},*}\right)}\right)^{-1}. \end{cases} \quad (15)$$

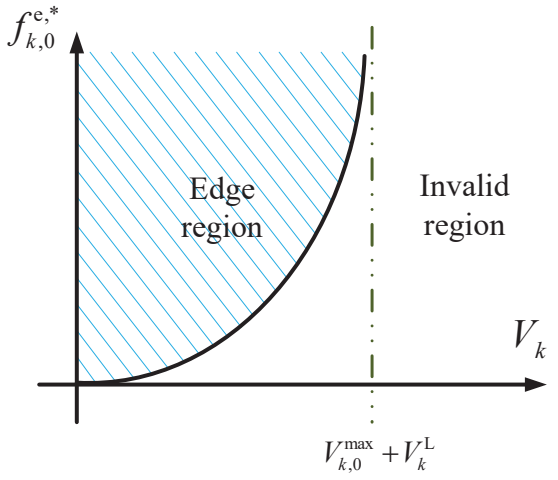
Case 3: If $V_{k,i}^{\max} + V_k^{\mathrm{L}} \le V_k$, there is no feasible solution.

Proof: Please see Appendix C. ■

Remark 1: The first case happens when the task data size is small, i.e., $V_k \le V_{k,i}^{\mathrm{D}}\left(p_{k,i}^{\mathrm{D,max}}\right) + V_k^{\mathrm{L}}$. In such a case, the task can be completed with only local computing and D2D offloading without edge computing. Therefore, $f_{k,i}^{\mathrm{e},*} = 0$, the device power should be totally allocated for D2D offloading, and no edge computation resource should be allocated to the corresponding device. In the second case, the D2D offloading is not enough. Therefore, the task should be offloaded to both the D2D device and the edge cloud. Correspondingly, the transmit power should be properly allocated to them. In this case, the required edge computation resource can be expressed in a closed-form manner, as presented in (15). The last case happens when the size of data is too large to be processed on time even if the overall edge computation resource at the BS is fully utilized by the device k. Therefore, there is no feasible solution in this case. In this situation, we will then delete device i from $\mathcal{U}_k$, delete device k from $\mathcal{U}_i$, and set $f_{k,i}^{\mathrm{e},*} = \infty$ before solving $\mathcal{P}3$.



(a) The general solution to $\mathcal{P}2$.



(b) The optimal solution for special case.

Fig. 4. The illustration of the optimal solution to $\mathcal{P}2$. The curve line represents the optimal value for different data sizes and the area filled with oblique lines represents the feasible region.

To be more specific, we depict the optimal edge computation resource allocation with different values of V_k, as shown in Fig. 4(a). Note that the D2D region and the cooperation region correspond to the case 1 and case 2, respectively.

B. A Special Case

To demonstrate the advantages by combing the D2D communications and MEC, we further investigate a special case where device k can only process its task by local computing and edge offloading without D2D offloading, i.e., $u_{k,0} = 1$.

With this consideration, the original problem can be simplified and the optimal solution to this case can be obtained, as summarized in the following.

Case 1: If $V_k < V_{k,0}^{\max} + V_k^{\mathrm{L}}$, the optimal solution is given by

$$\begin{cases} p_k^{\mathrm{C},*} = p_k^{\max}, \\ f_{k,0}^{\mathrm{e},*} = C_k \left(\frac{T}{V_k - V_k^{\mathrm{L}}} - \frac{1}{r_k^{\mathrm{C}}\left(p_k^{\max}\right)} \right). \end{cases} \tag{16}$$

Here, $V_{k,0}^{\max}$ is a special case of $V_{k,i}^{\max}$, i.e., $V_{k,0}^{\max} = Tr_k^{\mathrm{C}}\left(p_k^{\max}\right)$.

Case 2: If $V_k \geq V_{k,0}^{\max} + V_k^{\mathrm{L}}$, there is no feasible solution.[1]

Correspondingly, Fig. 4(b) depicts the optimal required edge computation resource, i.e., $f_{k,0}^{\mathrm{e},*}$, for the two cases. Similarly, there exists an invalid region when the task data size is too large.

Based on Theorem 2 and the above results, we can discuss the advantages benefiting from D2D offloading, which is elaborated in the following lemma.

Lemma 2: The threshold of invalid region in Fig. 4(a), i.e., $V_{k,i}^{\max} + V_k^{\mathrm{L}}$, is no less than the one in Fig. 4(b), i.e., $V_{k,0}^{\max} + V_k^{\mathrm{L}}$.

Proof: Since $V_{k,0}^{\max}$ is a special case of $V_{k,i}^{\max}$ that is given by maximizing the following function

$$\phi_{k,i}\left(p^{\mathrm{D}}\right) = \left(\left(\frac{1}{r_{k,i}^{\mathrm{D}}\left(p^{\mathrm{D}}\right)} + \frac{C_k}{f_i} \right)^{-1} + r_k^{\mathrm{C}}\left(p_k^{\max} - p^{\mathrm{D}}\right) \right) T, \tag{17}$$

where $p^{\mathrm{D}} \in \left[0, p_{k,i}^{\mathrm{D,max}}\right]$ and $\phi_{k,i}(0)$ equals to $V_{k,0}^{\max}$, we have

$$V_{k,i}^{\max} = \max_{p^{\mathrm{D}} \in \left[0, p_{k,i}^{\mathrm{D,max}}\right]} \phi_{k,i}\left(p^{\mathrm{D}}\right) \geq \phi_{k,i}(0) = V_{k,0}^{\max}. \tag{18}$$

Hence, $V_{k,i}^{\max} + V_k^{\mathrm{L}}$ is no less than $V_{k,0}^{\max} + V_k^{\mathrm{L}}$. ■

Remark 2: From Lemma 2, the threshold of the invalid part without D2D offloading is smaller than the one with D2D offloading. This result indicates that certain computation capacity gain can be achieved by exploiting the D2D offloading, i.e., more computation tasks can be completed on time, which demonstrates the advantages of the proposed D2D-MEC scheme. Note that the D2D communications have already been demonstrated effectiveness in providing proximity gain, reuse gain, and hop gain [22]. In this work, we further exploit the D2D communications to improve the computation capacity of the mobile edge computing system.

V. Computation Capacity Maximization

In the above, we have developed the optimal task offloading and power allocation strategy to minimize the required edge computation resource for a given D2D pair. Based on this, we will further investigate the computation capacity maximization problem in this section. We will first develop an effective algorithm to solve $\mathcal{P}3$ and then discuss the upper and lower bounds for this algorithm.

[1]If there is no feasible solution to $\mathcal{P}2$ in this special case, we will set $u_{k,0} = 0$ and $f_{k,0}^{\mathrm{e},*} = \infty$.

A. The Optimal Solution

Note that $\mathcal{P}3$ is still not easy to solve due to the complicated relationship between π_k and $u_{k,i}$. In what follows, we first transform it into a better tractable form and then develop an effective solution based on exhaustive search. Recall that the goal of $\mathcal{P}3$ is to maximize the number of supported devices under certain constraints on communication and computation resources. Thus, we assume that there are J $(0 \leq J \leq K)$ devices that can be supported and then test whether the communication and computation resource constraints can be fulfilled by solving the following optimization problem

$$\mathcal{P}4: \min_{\{u_{k,i}\}} \quad F_J^{\mathrm{e}} = \sum_{k=1}^{K} \sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i} f_{k,i}^{\mathrm{e},*}, \tag{19a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i} = J, \tag{19b}$$

$$\sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i} \leq 1, \ \forall k \in \mathcal{K}, \tag{19c}$$

$$\sum_{k \in \mathcal{U}_i} u_{k,i} \leq 1, \ \forall i \in \mathcal{I}, \tag{19d}$$

$$u_{k,i} \in \{0,1\}, \ \forall i \in \mathcal{U}_k \cup \{0\}, \forall k \in \mathcal{K}. \tag{19e}$$

Denote $F_J^{\mathrm{e},*}$ as the optimal value of $\mathcal{P}4$. Then, we can test the feasibility of the solution by comparing $F_J^{\mathrm{e},*}$ with the total edge computation resource, i.e., F^{e}. That is, if $F^{\mathrm{e}} \leq F_J^{\mathrm{e},*}$, the edge computation resource is adequate so that all J devices can be supported. Otherwise, this solution is infeasible. Therefore, we can achieve the optimal solution to $\mathcal{P}3$ by exhaustively searching J from K to 0 until the optimal value of $\mathcal{P}4$ with respect to J is feasible.

To solve $\mathcal{P}4$, we relax the integer variables, $u_{k,i}$, into continuous variables, and obtain the following problem

$$\mathcal{P}5: \min_{\{u_{k,i}\}} \quad F_J^{\mathrm{e}} = \sum_{k=1}^{K} \sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i} f_{k,i}^{\mathrm{e},*}, \tag{20a}$$

$$\text{s.t.} \quad \text{(19b), (19c), and (19d)},$$

$$0 \leq u_{k,i} \leq 1, \ \forall i \in \mathcal{U}_k \cup \{0\}, \forall k \in \mathcal{K}, \tag{20b}$$

which is a linear programming problem and can be solved by the Simplex Method [23] or Karmarkar's algorithm [24].

Although the integer variables are relaxed, we have the following theorem to show the equivalence between $\mathcal{P}4$ and $\mathcal{P}5$.

Theorem 3: The optimal solution to $\mathcal{P}5$, denoted by $u_{k,i}^*$, is either 1 or 0. Therefore, $\mathcal{P}5$ and $\mathcal{P}4$ have the same optimal solution.

Proof: Please see Appendix D. ■

B. Upper and Lower Bounds

Thus far, we have developed an effective algorithm to solve $\mathcal{P}3$. However, the optimal solution is based on exhaustive searching from K to 0, which is very time consuming. In the following, we aim to derive the upper and lower bounds of the optimal J to narrow the searching space.

Recall that one important restriction for enhancing the system computation capacity, i.e., the maximum number of

supported devices, is the limited edge computation resource. If we assume that the edge computation resource is infinite, the corresponding resource constraint (8b) can be removed from $\mathcal{P}3$, leading to the new problem as follows

$$\mathcal{P}6: \quad \max_{\{u_{k,i},\pi_k\}} \quad \sum_{k=1}^{K} \pi_k, \tag{21a}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{U}_k} u_{k,i} \le 1, \ \forall k \in \mathcal{K}, \tag{21b}$$

$$\sum_{k\in\mathcal{U}_i} u_{k,i} \le 1, \ \forall i \in \mathcal{I}, \tag{21c}$$

$$u_{k,i}, \pi_k \in \{0,1\}, \ \forall i \in \mathcal{U}_k \cup \{0\}, \ \forall k \in \mathcal{K}. \tag{21d}$$

By solving the above problem, we can derive an upper bound for J, denoted as J_{u}. We shall note that J_{u} is not necessarily equal to K since some devices might not complete their tasks on time due to their poor channel quality or large data size, even if the computation resource at the edge cloud is adequate. The above problem can be easily transformed to an assignment problem with the cost matrix, as

$$\mathbf{M} = \left[\begin{array}{ccc:ccc} c_{1,1}^{(1)} & \cdots & c_{1,K}^{(1)} & c_{1,1}^{(2)} & \cdots & c_{1,I}^{(2)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{K,1}^{(1)} & \cdots & c_{K,K}^{(1)} & c_{K,1}^{(2)} & \cdots & c_{K,I}^{(2)} \end{array}\right]. \tag{22}$$

The elements in $\mathbf{M}$ are given as: $c_{k_1,k_2}^{(1)}$ equals 1 if $k_1 = k_2$ and $f_{k_1,0}^{\mathrm{e},*} < \infty$, and ∞ otherwise, $c_{k,i}^{(2)}$ equals 1 if $i \in \mathcal{U}_k$ and $f_{k,i}^{\mathrm{e},*} < \infty$, and ∞ otherwise. Therefore, the classical Hungarian algorithm [25] can be applied to obtain the optimal pairs (k, n_k) of $\mathbf{M}$, with a computational complexity of $\mathcal{O}\left(N^3\right)$. Then, by only counting the pair whose corresponding element is less than ∞, we can obtain the upper bound.

Regarding the lower bound of J, we can assume that there is no edge computation resource in $\mathcal{P}3$, i.e., $\sum_{k=1}^{K}\sum_{i\in\mathcal{U}_k\cup\{0\}} u_{k,i} f_{k,i}^{\mathrm{e},*} = 0$. In this situation, we should set the elements in $\mathbf{M}$ in the following way. $c_{k_1,k_2}^{(1)} = \infty, \forall k_1, k_2 \in \mathcal{K}$. $c_{k,i}^{(2)}$ equals 1 if $i \in \mathcal{U}_k$ and $f_{k,i}^{\mathrm{e},*} = 0$, and ∞ otherwise. Then, the lower bound for J, denoted by J_{l}, can be achieved in a similar way as J_{u}.

C. The Optimal Algorithm and Discussion

Till now, we have optimally solved $\mathcal{P}1$ and also obtained the upper bound J_{u} and the lower bound J_{l} for the optimal value. The proposed algorithm first obtain $f_{k,i}^{\mathrm{e},*}$ for each D2D pair (k, i) and J_{u} and J_{l}. After that, the algorithm utilizes the bisection method to search the optimal value, J^*, from J_{l} to J_{u}, as presented in Algorithm 1. In each iteration, feasibility test should be conducted by solving $\mathcal{P}4$.

Regarding the computation complexity of the algorithm, the required iteration number in the bisection searching method can be described by $\mathcal{O}\left(\ln\left(J_{\mathrm{u}} - J_{\mathrm{l}}\right)\right)$. $\mathcal{P}4$ can be converted to $\mathcal{P}5$ and $\mathcal{P}5$ is then solved by the Karmarkar's algorithm with the complexity of $\mathcal{O}\left((KI)^{3.5} L \ln L \ln\ln L\right)$, where L is the number of input bits [24]. In $\mathcal{P}2$, $f_{k,i}^{\mathrm{e},*}$ can be solved by the closed-form expressions presented in Theorem 2, whose complexity can be neglected. Therefore, the computational complexity of the whole algorithm is $\mathcal{O}\left(\ln\left(J_{\mathrm{u}} - J_{\mathrm{l}}\right)(KI)^{3.5} L \ln L \ln\ln L\right)$, which is polynomial.

Algorithm 1 The optimal algorithm to $\mathcal{P}1$.

```
Obtain K, I according to devices' workload and computation resources.
Obtain U_k, U_i according to devices' locations.
Initialize f_{k,i}^{e,*}, ∀i ∈ U_k ∪ {0}, ∀k ∈ K.
Calculate J_u by solving P6.
Calculate J_l by solving P6 with Σ_{k=1}^{K} Σ_{i∈U_k∪{0}} u_{k,i} f_{k,i}^{e,*} = 0.
Calculate F_{J_u}^{e,*} by solving P4 with J = J_u.
if F^e ≥ F_{J_u}^{e,*} then
    return J_u.
else if F^e = 0 then
    return J_l.
else
    while J_u − J_l > 1 do
        J = ⌊(J_l + J_u)/2⌋.
        Calculate F_J^{e,*} by solving P4.
        if F_J^{e,*} > F^e then
            J_u = J.
        else
            J_l = J.
        end if
    end while
    return J_l.
end if
```

In the following, we provide some interesting insights of the proposed algorithm by analyzing some special cases. The first special case is that no edge computing is allowed and the task can be jointly computed locally and by a D2D device. In this case, the computation capacity of the system, i.e., the maximum number of supported devices, is exactly J_{l} as discussed before. Another special case corresponds to that D2D offloading is not allowed and each task can be jointly computed locally and by the edge cloud. In this case, the minimal required edge computation resource for each device is given by (16). We can further rearrange $f_{k,0}^{\mathrm{e},*}$ in the non-descend order, i.e., $f_{\tau(1),0}^{\mathrm{e},*} \le f_{\tau(2),0}^{\mathrm{e},*} \le \cdots \le f_{\tau(K),0}^{\mathrm{e},*}$ where $\boldsymbol{\tau}$ is a permutation on $\{1, 2, \cdots, K\}$. Then, we can find out the maximum J_s that satisfies

$$\sum_{k=1}^{J_s} f_{\tau(k),0}^{\mathrm{e},*} \le F^{\mathrm{e}}. \tag{23}$$

By this means, J_s is the maximum number of supported devices without the D2D offloading. Apparently, $J^* \ge J_s$ and $J^* \ge J_{\mathrm{l}}$. We introduce the following two gains related to our proposed algorithm.

- The D2D gain is defined by $J^* - J_s$, which represents the additional number of supported devices if tasks can be offloaded to D2D devices as compared to local computing and edge offloading.
- The MEC gain is defined by $J^* - J_{\mathrm{l}}$, which represents the additional number of supported devices if tasks can

be offloaded to the edge cloud as compared to local computing and D2D offloading.

In the next section, we will present more details on the D2D gain and MEC gain that can be achieved by the proposed algorithm through numerical simulation.

VI. Simulation Results

In this section, we will present simulation results to verify the performance enhancement of the proposed D2D-MEC system. The simulation settings are as follows unless otherwise stated. The BS has a radius of 300 m and 80 devices are randomly located in the coverage. Each device can establish one cellular link with the BS and one D2D link with a nearby device. The maximum range of each D2D link, i.e., R, is set as 50 m. The channel gains of all links are generated according to independent and identically distributed (i.i.d.) Rayleigh random variables with unit variances. The maximum transmit power is set as 24 dBm for all devices. The delay tolerance of all tasks is 1 s. The computation resource of each device, i.e., f_n, is uniformly generated from 0.5×10^9 to 2×10^9 CPU cycles/s. For the computation task, both the data size and the required number of CPU cycles per bit follow the uniform distribution within $V_n \in [0.1, 4]$ Mbits and $C_n \in [500, 1500]$ CPU cycles/s, respectively. All random variables are independent for different devices, modelling heterogeneous mobile computation resource. The total computation resource of the edge cloud is set as 40×10^9 CPU cycles/s. Table I summarizes the major simulation parameters.

TABLE I
Simulation Parameters

Parameter	Value
Cell radius	300 m
D2D range, R	50 m
Total number of devices, N	80
Delay tolerance, T	1 s
Sub-channel bandwidth, B	0.5 MHz
Noise spectral density	-174 dBm/Hz
Transmit power, $p_k^{\max}$	24 dBm
Computation resource of the device, f_n	$[0.5, 2] \times 10^9$ CPU cycles/s
Data size of task, V_n	$[0.1, 4]$ Mbits
Required number of CPU cycles per bit, C_n	$[500, 1500]$ CPU cycles/bit
Edge computation resource, F^{e}	40×10^9 CPU cycles/s

A. Performance Comparison

We first compare the performance of the proposed D2D-MEC system with two benchmark systems.

- the *MEC system* where each task can be collaboratively processed with only local computing and edge computing [6]. Note that the number of supported devices in this system is given by J_s as discussed before.
- the *D2D system* where each task can be collaboratively processed with only local computing and D2D offloading [19]. Note that the number of supported devices in this system is given by J_{l} as discussed before.

By comparing J_s and J_{l} with the maximized device number in our algorithm, i.e., J^*, both D2D gain and MEC gain can be observed.

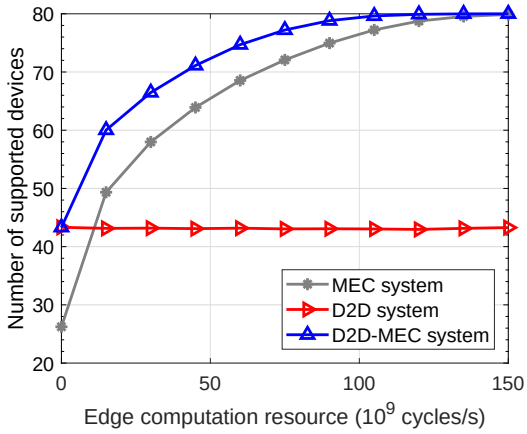


Fig. 5. The number of supported devices vs. the edge computation resource.

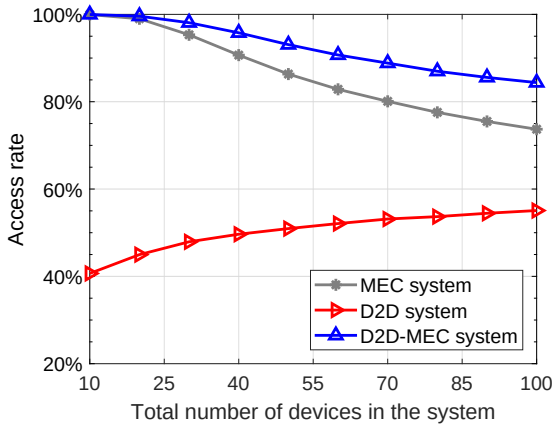


Fig. 6. Access rate vs. the total number of devices in the system.

Fig. 5 depicts the number of supported devices versus the computation resource of the edge cloud in the three different systems. First, the numbers of supported devices in both the MEC system and the D2D-MEC system increase with the edge computation resource, while the total number of supported devices in the D2D system keeps invariant since the D2D system does not utilize the edge computation resource. Secondly, the D2D-MEC system always achieves the best performance among the three systems. When the edge computation resource is insufficient, the D2D-MEC system can evidently outperform the MEC system since the D2D-MEC system can fully utilize computation resources of mobile devices and the edge cloud. Moreover, the computation capacity of the D2D-MEC system is larger than the MEC system since the computation resource of the devices in $\mathcal{I}$ can be used for processing the offloaded tasks. Finally, the upper bounds of the computation capacities of the D2D-MEC system and the MEC system are the same, that is the overall number of devices, i.e., 80. However, the overall required edge computation resources for the D2D-MEC system and MEC system to support 80 devices are 113.8×10^9 CPU cycles/s and 148.4×10^9 CPU cycles/s, respectively. This result reveals that exploiting D2D communications can signif-

icantly save the overall required edge computation resource. From Fig. 5, we can clearly observe the D2D gain and MEC gain that are achieved by our proposed algorithm.

Fig. 6 shows the access rate, i.e., the proportion of devices that can be supported versus the total number of devices. It can be observed that the access rates of both the MEC system and the D2D-MEC system decrease with the number of devices due to the limited edge computation resource, while the access rate of the D2D system will correspondingly increase. The reason is that, the allocated edge computation resource for each device in the MEC system and the D2D-MEC system decreases with the total number of devices, which results in the smaller proportion of devices that can be supported. Furthermore, the device computation resource can be fully utilized as the total number of devices in the D2D system increases, which eventually improves the access rate. We can also observe that the access rate of the D2D-MEC system decreases slower than the MEC system since it utilizes the device computation resource for capacity enhancement.

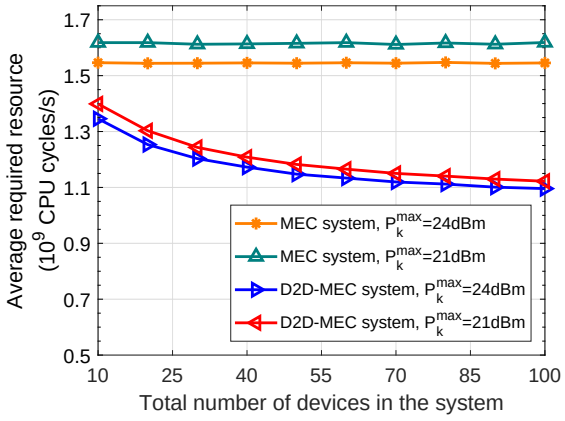


Fig. 7. The average required edge computation resource vs. the total number of devices in the system.

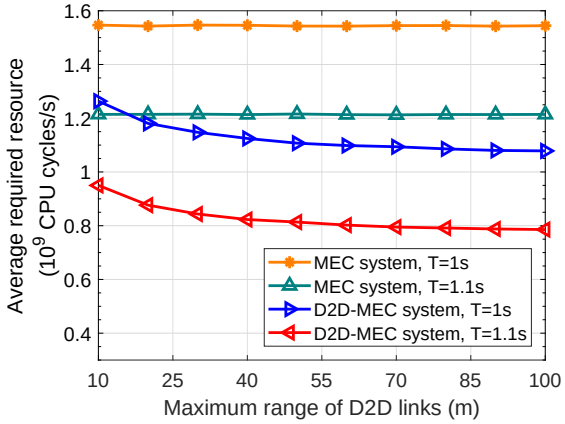


Fig. 8. The average required edge computation resource vs. the maximum range of D2D links.

Fig. 7 shows the average required edge computation resource for each device versus the total number of devices in the MEC system and the D2D-MEC system, respectively. It can be observed that the average required edge computation resource of the D2D-MEC system decreases with the total number of devices. The reason can be explained as follows. With more devices in the system, each device would have more opportunities to offload data to a nearby device, which reduces the required computation resource at the edge cloud. However, the average required edge computation resource of the MEC system keeps almost unchanged since D2D offloading is not exploited. Moreover, the D2D-MEC system always requires less computation resource at the edge cloud than the MEC system. Besides, with larger transmit power, the average required edge computation resources in both the MEC system and the D2D-MEC system become smaller because of the shorter transmission delay.

Fig. 8 depicts the average required edge computation resource for each device versus the maximum range of D2D links in the MEC system and the D2D-MEC system, respectively. It can be seen that the average required computation resource of the D2D-MEC system is smaller than that in the MEC system, especially when the maximum range of D2D links becomes large. The reason is that, each device has more potential nearby devices to offload data as the range of D2D link becomes larger. In this way, the computation resources of all devices can be further utilized via D2D offloading. From the figure, we can also observe a reduced required computation resource with the decreased delay tolerance in both systems.

B. Impact of the System Parameters

Next, we analyze the impact of the system parameters, i.e., the edge computation resource and the total number of device, in the D2D-MEC system. For easy discussion, we first classify the supported devices in the D2D-MEC system into the following four kinds:

- *Local devices*, which complete their tasks with local computing only, i.e., those devices in $\mathcal{I}$.
- *D2D devices*, which complete their tasks with local computing and D2D offloading.
- *Edge devices*, which complete their tasks with local computing and edge offloading.
- *Cooperation devices*, which complete their tasks with local computing, edge offloading, and D2D offloading.

Fig. 9 illustrates the impact of the edge computation resource on the number of devices in each kind. From the figure, both the numbers of edge devices and cooperation devices increase with edge computation resource. The reason is that more devices can complete their tasks with the assistance of more powerful edge cloud. Hence, the number of overall supported devices also increases with the edge computing resource. However, the number of D2D devices decreases with the edge computation resource because each device is inclined to offload its task to the more powerful edge cloud rather than the less powerful nearby device.

Fig. 10 presents the access rate versus the total number of devices in the system. Due to the limitation on the edge computation resource, the access rate decreases with the total number of devices. It is interesting that the access rate of cooperation devices first increases and then decreases with the total number of devices. The reason can be explained as

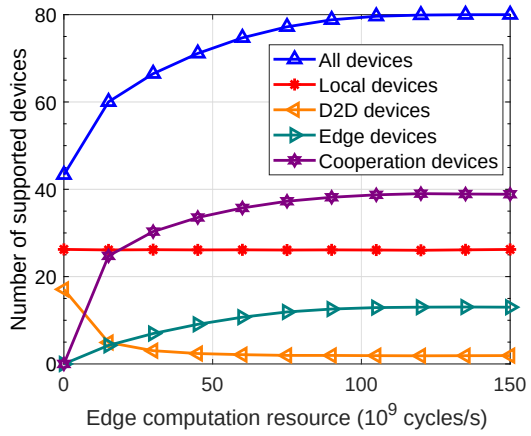


Fig. 9. The number of supported devices vs. the edge computation resource of different kinds.

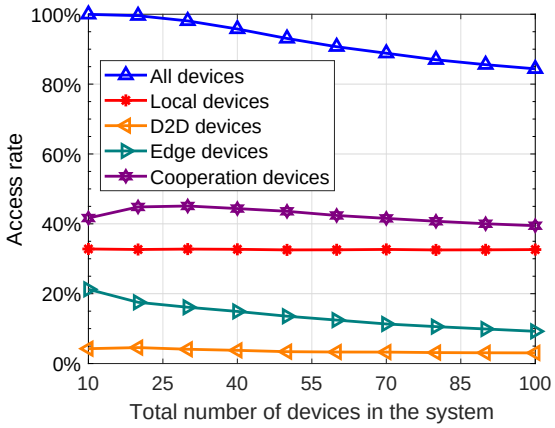


Fig. 10. Access rate vs. the total number of devices of different kinds.

follows. When the number of devices is small, the edge computation resource is adequate to support almost all devices in the system, leading to the increasing access rate of cooperation devices. On the contrary, the access rate decreases with the total number of devices due to the limitation on the total edge computation resource when the number of devices becomes large.

VII. Conclusion

This paper proposes a multi-user D2D-MEC system to improve the computation capacity of the whole system, where each task can be simultaneously offloaded for both edge computing and D2D computing. A mixed integer non-linear problem to maximize the system computation capacity is first formulated. Then, we decomposed it into two subproblems and proved that the optimal solutions of them also compose the optimal one to the original problem. Specifically, the first subproblem is to minimize the required edge computation resources for a given D2D pair. By solving it, the optimal transmit power allocation and data offloading strategy can be derived in closed-form by leveraging the KKT conditions. Based on the solution to the first subproblem, the second subproblem maximizes the computation capacity of the D2D-MEC system. We then developed an effective algorithm to achieve the optimal solution to the second subproblem based on exhaustive searching. Lower and upper bounds of the optimal value are also derived to narrow the searching space as well as some interesting insights are revealed. Finally, numerical simulation results demonstrate that the proposed algorithm can effectively improve the system computation capacity as compared with some benchmark systems.

Our initial study in this work have demonstrated the potential of applying D2D communications to further enhance the computation capacity of a cellular network. To gain more insightful results, we have made some assumptions on the system model. In our future work, we will further develop some practical techniques to facilitate the implementation of the proposed D2D-MEC system. First, we have assumed that each D2D pair is allocated with one orthogonal sub-channel. In the future, sub-channel reuse can be considered to further improve the system performance, where joint computation offloading, resource allocation, and interference management should be considered. Second, the objective function in $\mathcal{P}1$ may lead to unfairness among devices. Therefore, we can further develop some fair resource allocation algorithms based on the max-min criterion [26]. Third, we may also investigate the incentive mechanism in the D2D-MEC system, where utility function and game theoretical techniques can be used to analyze the actions of devices and the BS. Last but not the least, another interesting issue is to study the energy efficiency of the D2D-MEC system, i.e., minimizing the energy consumption under communication and computation resource limitations.

Appendix A
Proof of Theorem 1

With the fact that both $\mathcal{P}1$ and $\mathcal{P}3$ have the same objective function, we can prove that they have the same optimal solution by proving that the optimal solution to $\mathcal{P}1$ is feasible to $\mathcal{P}3$, and vice versa. To prove the optimal solution to $\mathcal{P}1$ ($\mathcal{P}3$) is a feasible solution to $\mathcal{P}3$ ($\mathcal{P}1$), we need to check whether all the constraints in $\mathcal{P}3$ ($\mathcal{P}1$) can be satisfied by the optimal solution to $\mathcal{P}1$ ($\mathcal{P}3$). Based on this, we give the detailed proof as follows.

Let $\left\{\pi_k^{*,1}, u_{k,i}^{*,1}, f_{k,i}^{\mathrm{e},*,1}\right\}$ denote the optimal solution to $\mathcal{P}1$ and $J^{*,1} = \sum_{k=1}^{K} \pi_k^{*,1}$ denote the optimal value. Let $f_{k,i}^{\mathrm{e},*,2}$ denote the optimal value to $\mathcal{P}2$. Let $\left\{\pi_k^{*,3}, u_{k,i}^{*,3}\right\}$ denote the optimal solution to $\mathcal{P}3$ and $J^{*,3} = \sum_{k=1}^{K} \pi_k^{*,3}$ denote the optimal value. Since $\mathcal{P}2$ aims to minimize the required edge computation resource of device k for a given D2D pair (k,i), $f_{k,i}^{\mathrm{e},*,2}$ must be no more than $f_{k,i}^{\mathrm{e},*,1}$, i.e., $f_{k,i}^{\mathrm{e},*,2} \le f_{k,i}^{\mathrm{e},*,1}$. Then, we have

$$\sum_{k=1}^{K} \sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i}^{*,1} f_{k,i}^{\mathrm{e},*,2} \le \sum_{k=1}^{K} \sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i}^{*,1} f_{k,i}^{\mathrm{e},*,1} \le F^{\mathrm{e}}, \quad (24)$$

which means that $u_{k,i}^{*,1}$ satisfies constraint (8b). Besides, $\left\{u_{k,i}^{*,1}, \pi_k^{*,1}\right\}$ also satisfies other constraints of $\mathcal{P}3$, therefore, it is a feasible solution to $\mathcal{P}3$. Hence, $J^{*,1}$ is no more than $J^{*,3}$, i.e., $J^{*,1} \le J^{*,3}$.

On the other hand, the optimal value of $\mathcal{P}2$, $f_{k,i}^{\mathrm{e},*,2}$, satisfies (6b). The optimal solution to $\mathcal{P}3$, $\left\{\pi_k^{*,3}, u_{k,i}^{*,3}\right\}$, also

satisfies constraints (6f) and (6g). Moreover, since $f_{k,i}^{\mathrm{e},*,2}$ satisfies constraint (6b) and $\left\{f_{k,i}^{\mathrm{e},*,2}, u_{k,i}^{*,3}\right\}$ satisfies constraint (8b), $\left\{f_{k,i}^{\mathrm{e},*,2}, u_{k,i}^{*,3}\right\}$ also satisfies constraint (6e) with $f_k^{\mathrm{e}} = \sum_{i\in\mathcal{U}_k\cup\{0\}} u_{k,i}^{*,3} f_{k,i}^{\mathrm{e},*,2}$. As for remaining constraints of $\mathcal{P}1$, it is easy to find they are also satisfied. Therefore, $\left\{u_{k,i}^{*,3}, \pi_k^{*,3}, f_{k,i}^{\mathrm{e},*,2}\right\}$ is a feasible solution to $\mathcal{P}1$. Hence, $J^{*,1}$ is no less than $J^{*,3}$, i.e., $J^{*,1} \geq J^{*,3}$.

In conclusion, the optimal value of $\mathcal{P}3$ equals the optimal value of $\mathcal{P}1$, which ends the proof.

Appendix B
Proof of Lemma 1

It is obvious that the objective function and the constraints (7d) - (7g) of $\mathcal{P}2$ are convex. Hence, we focus on the constraints (7b) and (7c). First, we define a new function as

$$f(x,y) = -\left(\frac{C_1}{\ln(1+x)} + \frac{C_2}{y}\right)^{-1}, \tag{25}$$

where $x, y, C_1, C_2 > 0$. The Hessian of $f(x,y)$ is

$$\boldsymbol{H} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial y \partial x} & \dfrac{\partial^2 f}{\partial y^2} \end{bmatrix}. \tag{26}$$

The leading principal mirrors of $\boldsymbol{H}$ are given by

$$\triangle_1 = \frac{C_1 y^2 \left(2C_2 + C_1 y + C_2 \ln(1+x)\right)}{(1+x)^2 \left(C_1 y + C_2 \ln(1+x)\right)^3} > 0, \tag{27}$$

$$\triangle_2 = \frac{2C_1^2 C_2 y^2 \ln^2(1+x)}{(1+x)^2 \left(C_1 y + C_2 \ln(1+x)\right)^5} > 0. \tag{28}$$

From the above analysis, $f(x,y)$ is convex and so are the constraints (7b) and (7c). Therefore, $\mathcal{P}2$ is a convex optimization problem. This ends the proof.

Appendix C
Proof of Theorem 2

Since $\mathcal{P}2$ is convex and the partial Lagrangian function has been given by (9), the necessary and sufficient conditions based on the KKT conditions can be expressed as

$$\frac{\partial L}{\partial f_{k,i}^{\mathrm{e},*}} = 1 - \lambda^* \frac{TC_k \left(r_k^{\mathrm{C}}\left(p_k^{\mathrm{C},*}\right)\right)^2}{\left(f_{k,i}^{\mathrm{e},*} + C_k r_k^{\mathrm{C}}\left(p_k^{\mathrm{C},*}\right)\right)^2} \begin{cases} = 0, & f_{k,i}^{\mathrm{e},*} > 0, \\ \geq 0, & f_{k,i}^{\mathrm{e},*} = 0, \end{cases} \tag{29}$$

$$\frac{\partial L}{\partial p_k^{\mathrm{D},*}} = \omega^* - \frac{\mu^* f_i^2 T}{\left(f_i + C_k r_{k,i}^{\mathrm{D}}\left(p_k^{\mathrm{D},*}\right)\right)^2} \frac{\mathrm{d}r_k^{\mathrm{D}}\left(p_k^{\mathrm{D}}\right)}{\mathrm{d}p_k^{\mathrm{D}}}\Big|_{p_k^{\mathrm{D}}=p_k^{\mathrm{D},*}} \begin{cases} = 0, & p_k^{\mathrm{D},*} > 0, \\ \geq 0, & p_k^{\mathrm{D},*} = 0, \end{cases} \tag{30}$$

$$\frac{\partial L}{\partial p_k^{\mathrm{C},*}} = \omega^* - \frac{\lambda^* \left(f_{k,i}^{\mathrm{e},*}\right)^2 T}{\left(f_{k,i}^{\mathrm{e},*} + C_k r_k^{\mathrm{C}}\left(p_k^{\mathrm{C},*}\right)\right)^2} \frac{\mathrm{d}r_{k,i}^{\mathrm{C}}\left(p_k^{\mathrm{C}}\right)}{\mathrm{d}p_k^{\mathrm{C}}}\Big|_{p_k^{\mathrm{C}}=p_k^{\mathrm{C},*}} \begin{cases} = 0, & p_k^{\mathrm{C},*} > 0, \\ \geq 0, & p_k^{\mathrm{C},*} = 0, \end{cases} \tag{31}$$

$$\frac{\partial L}{\partial V_k^{\mathrm{D},*}} = -\lambda^* + \mu^* \begin{cases} \leq 0, \ V_k^{\mathrm{D},*} = \min\left(\frac{Tf_i - V_i C_i}{C_k}, V_k - V_k^{\mathrm{L}}\right), \\ = 0, \ 0 < V_k^{\mathrm{D},*} < \min\left(\frac{Tf_i - V_i C_i}{C_k}, V_k - V_k^{\mathrm{L}}\right), \\ \geq 0, \ V_k^{\mathrm{D},*} = 0. \end{cases} \tag{32}$$

$$\lambda^* \left(\left(V_k - V_k^{\mathrm{D},*} - V_k^{\mathrm{L}}\right) - T\left(\frac{1}{r_k^{\mathrm{C}}(p_k^{\mathrm{C},*})} + \frac{C_k}{f_{k,i}^{\mathrm{e},*}}\right)^{-1}\right) = 0, \tag{33}$$

$$\mu^* \left(V_k^{\mathrm{D},*} - T\left(\frac{1}{r_{k,i}^{\mathrm{D}}(p_k^{\mathrm{D},*})} + \frac{C_k}{f_i}\right)^{-1}\right) = 0, \tag{34}$$

$$\omega^* \left(p_k^{\mathrm{D},*} + p_k^{\mathrm{C},*} - p_k^{\max}\right) = 0, \tag{35}$$

$$\lambda^*, \mu^*, \omega^* \geq 0. \tag{36}$$

Then, we can discuss the solution based on the value of $f_{k,i}^{\mathrm{e},*}$ as follows.

Case 1: If $f_{k,i}^{\mathrm{e},*} = 0$, it means that edge offloading is not necessary and we can allocate all transmit power to D2D transmission, i.e., $p_k^{\mathrm{D},*} = p_k^{\max}$, $p_k^{\mathrm{C},*} = 0$, and $V_k^{\mathrm{D},*} = V_k - V_k^{\mathrm{L}}$. Meanwhile, the other constraints should be satisfied, meaning that this case happens when $V_k^{\mathrm{D},*}$ satisfies

$$V_k^{\mathrm{D},*} = V_k - V_k^{\mathrm{L}} \leq \min\left(T\left(\frac{1}{r_{k,i}^{\mathrm{D}}(p_k^{\max})} + \frac{C_k}{f_i}\right)^{-1}, \frac{Tf_i - V_i C_i}{C_k}\right). \tag{37}$$

Since we have defined the maximal D2D transmit power in (10), and the D2D data offloading size function in (11), the above condition can be simplified as

$$V_k \leq V_{k,i}^{\mathrm{D}}\left(p_{k,i}^{\mathrm{D,max}}\right) + V_k^{\mathrm{L}}. \tag{38}$$

Case 2: If $f_{k,i}^{\mathrm{e},*} > 0$, it means that both edge offloading and D2D offloading should be utilized. Then, according to (29) and (31), both λ^* and ω^* must be positive. We can first consider the situation when $\lambda^* = \mu^*$. Then, based on (30) and (31), we have

$$\frac{f_i^2 T}{\left(f_i + C_k r_{k,i}^{\mathrm{D}}\left(p_k^{\mathrm{D},*}\right)\right)^2} \frac{\mathrm{d}r_{k,i}^{\mathrm{D}}\left(p_k^{\mathrm{D}}\right)}{\mathrm{d}p_k^{\mathrm{D}}}\Big|_{p_k^{\mathrm{D}}=p_k^{\mathrm{D},*}} = \frac{\left(f_{k,i}^{\mathrm{e},*}\right)^2 T}{\left(f_{k,i}^{\mathrm{e},*} + C_k r_k^{\mathrm{C}}\left(p_k^{\mathrm{C},*}\right)\right)^2} \frac{\mathrm{d}r_k^{\mathrm{C}}\left(p_k^{\mathrm{C}}\right)}{\mathrm{d}p_k^{\mathrm{C}}}\Big|_{p_k^{\mathrm{C}}=p_k^{\mathrm{C},*}}. \tag{39}$$

Meanwhile, since $\lambda^*, \mu^*, \omega^* > 0$, constraints (7b), (7c), and (7e) should be equality constraints. Then, (39) can be rewritten as

$$\varphi_{k,i}\left(p_k^{\mathrm{D},*}\right) = 0, \tag{40}$$

where $\varphi_{k,i}\left(p^{\mathrm{D}}\right)$ is the transmit power allocation function given by (12). Then, we have $p_k^{\mathrm{D},*} = \Phi_{k,i}(0)$

and $V_k^{\mathrm{D},*} = V_{k,i}^{\mathrm{D}}\left(p_k^{\mathrm{D},*}\right)$, where $\Phi_{k,i}(x)$ is the inverse function of $\varphi_{k,i}\left(p_k^{\mathrm{D}}\right)$. We should note that $V_k^{\mathrm{D},*} \leq \min\left(\left(Tf_i - V_iC_i\right)/C_k, V_k - V_k^{\mathrm{L}}\right)$ according to (32). Therefore, $p_k^{\mathrm{D},*} \leq p_{k,i}^{\mathrm{D},\max}$ and $V_{k,i}^{\mathrm{D}}\left(p_k^{\mathrm{D},*}\right) \leq V_k - V_k^{\mathrm{L}}$, which is the condition of this situation. However, since $\varphi_{k,i}\left(p_k^{\mathrm{D},*}\right) = 0$ and $p_k^{\mathrm{D},*} \leq p_{k,i}^{\mathrm{D},\max}$ may not be satisfied at the same time, it is hard to express the condition and solution of this situation. Similarly, by analyzing other situations, i.e., $\lambda^* > \mu^*$ and $\lambda^* < \mu^*$, $p_k^{\mathrm{D},*}$ is given by

$$p_k^{\mathrm{D},*} = \min\left(\Phi_{k,i}(0), p_{k,i}^{\mathrm{D},\max}\right). \tag{41}$$

The corresponding condition is

$$V_{k,i}^{\mathrm{D}}\left(p_{k,i}^{\mathrm{D},\max}\right) + V_k^{\mathrm{L}} < V_k < V_{k,i}^{\max} + V_k^{\mathrm{L}}, \tag{42}$$

where $V_{k,i}^{\max}$ is the total offloading data size in (13) which gives the upper bound of the task size that edge offloading and D2D offloading can process.

Case 3: When the task size is beyond the total offloading data size, i.e., $V_k \geq V_{k,i}^{\max} + V_k^{\mathrm{L}}$, there is no feasible solution.

Appendix D
Proof of Theorem 3

First of all, $\mathcal{P}5$ can be rewritten as

$$\mathcal{P}7: \quad \min_{\{u_{k,i}\}} \quad F_J^{\mathrm{e}} = \sum_{k=1}^{K} \sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i} f_{k,i}^{\mathrm{e},*}, \tag{43a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} u_{k,-1} = K - J, \tag{43b}$$

$$\sum_{i \in \mathcal{U}_k \cup \{0\}} u_{k,i} + u_{k,-1} = 1, \ \forall k \in \mathcal{K}, \tag{43c}$$

$$\sum_{k \in \mathcal{U}_i} u_{k,i} + u_{-2,i} = 1, \ \forall i \in \mathcal{I}, \tag{43d}$$

$$u_{k,i} \geq 0, \ \forall i \in \mathcal{U}_k \cup \{0\}, \forall k \in \mathcal{K}, \tag{43e}$$

$$u_{k,-1}, u_{-2,i} \geq 0, \ \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \tag{43f}$$

In the above, we define two auxiliary nodes, namely node -1 and node -2. Let $u_{k,-1}$ characterize the connectivity between the node -1 and the device k in $\mathcal{K}$. Let $u_{-2,i}$ characterize the connectivity between the node -2 and the device i in $\mathcal{I}$. Then, we can use a bipartite graph $G = (U, V, E)$ to describe the problem, where $U = \mathcal{K} \cup \{-2\}, V = \mathcal{I} \cup \{-1, 0\}$, and E is the set of edges. Let $\boldsymbol{A}$ denote the incidence matrix of G and it is a totally unimodular matrix according to Chapter 19.3 in [27]. Meanwhile, $\mathcal{P}7$ can be expressed in the canonical form, as

$$\min\left\{\boldsymbol{f}^T\boldsymbol{u} | \boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}; \boldsymbol{u} \geq 0\right\}, \tag{44}$$

where $\boldsymbol{f}$ is a vector composed by $f_{k,i}^{\mathrm{e},*}$ and 0, $\boldsymbol{u}$ is a vector composed by $u_{k,i}$, and $\boldsymbol{b}$ is a vector that only contains 1 and $K - J$.

Following Theorem 19.1 in [27], the polyhedron $\{\boldsymbol{u} | \boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}, \boldsymbol{u} \geq 0\}$ has only integer vertices. Meanwhile, according to Theorem 13.3 in [28], all basic feasible points of this LP problem are vertices of the feasible polytope $\{\boldsymbol{u} | \boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}, \boldsymbol{u} \geq 0\}$, and vice versa. Combining these two theorems, we can prove that all the basic feasible points of this LP problem are integer vectors, which further indicates that the optimal solution is also an integer vector. Since $u_{k,i}$ belongs to $[0, 1]$, the optimal solution, i.e., $u_{k,i}^*$, must be either one or zero. Besides, according to Chapter 13.3 in [28], all iterates of the simplex method are basic feasible points, which indicates that we can use the simplex method to guarantee that the optimal solution is an integer vector even if the optimal solution is not unique.

Let OPT_4 and OPT_5 denote the optimal value of $\mathcal{P}4$ and $\mathcal{P}5$, respectively. Then we have $\mathrm{OPT}_5 \leq \mathrm{OPT}_4$. Meanwhile, the optimal solution to $\mathcal{P}5$, $u_{k,i}^*$, is also a feasible solution to $\mathcal{P}4$. Hence, $\mathrm{OPT}_5 = \mathrm{OPT}_4$. In this way, we have proved that the optimal value is unchanged even if we relax the integer variables $u_{k,i}$.

References

[1] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, Jul. 2015.

[2] European Telecommunications Standards Institute (ETSI), "Mobile edge computing-introductory technical white paper," Sep. 2014.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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)*, Jul. 2016, pp. 1451–1455.

[9] 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.

[10] 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.

[11] 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.

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

[13] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklós, and Z. Turányi, "Design aspects of network assisted device-to-device communications," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 170–177, Mar. 2012.

[14] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1801–1819, 4th Quart. 2014.

[15] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-device communication in LTE-advanced networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1923–1940, 4th Quart. 2015.

[16] Y. Li, L. Sun, and W. Wang, "Exploring device-to-device communication for mobile cloud computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 2239–2244.

[17] W. Hu and G. Cao, "Quality-aware traffic offloading in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3182–3195, Nov. 2017.

[18] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.
[19] C. You and K. Huang, "Exploiting non-causal CPU-state information for energy-efficient mobile cooperative computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4104–4117, Jun. 2018.
[20] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K.-K. Wong, R. Schober, and L. Hanzo, "User association in 5G networks: A survey and an outlook," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1018–1044, 2nd Quart. 2016.
[21] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu, "Exploiting massive D2D collaboration for energy-efficient mobile edge computing," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 64–71, Aug. 2017.
[22] G. Yu, L. Xu, D. Feng, R. Yin, G. Y. Li, and Y. Jiang, "Joint mode selection and resource allocation for device-to-device communications," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 3814–3824, Nov. 2014.
[23] D. Gale, "Linear programming and the simplex method," *Notices AMS*, vol. 54, no. 3, pp. 364–369, Mar. 2007.
[24] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proc. ACM STOC*, 1984, pp. 302–311.
[25] H. W. Kuhn, "The hungarian method for the assignment problem," *Nav. Res. Logist.*, vol. 2, no. 1-2, pp. 83–97, 1955.
[26] L. B. Le, "Fair resource allocation for device-to-device communications in wireless cellular networks," in *Proc. IEEE Global Commun. Conf.*, Anaheim, CA, Dec. 2012, pp. 5451–5456.
[27] A. Schrijver, *Theory of linear and integer programming*. John Wiley& Sons, 1998.
[28] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.

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.

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.

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.