Latency Minimization for D2D-Enabled Partial Computation Offloading in Mobile Edge Computing

Umber Saleem, Yu Liu, Sobia Jangsher, *Member, IEEE*, Xiaoming Tao, *Member, IEEE*, and Yong Li, *Senior Member, IEEE*

***Abstract*—We consider Device-to-Device (D2D)-enabled mobile edge computing offloading scenario, where a device can partially offload its computation task to the edge server or exploit the computation resources of proximal devices. Keeping in view the millisecond-scale latency requirement in 5G service scenarios and the spectrum scarcity, we focus on minimizing the sum of task execution latency of all the devices in a shared spectrum with interference. In particular, we provide an integrated framework for partial offloading and interference management using orthogonal frequency-division multiple access (OFDMA) scheme. Accordingly, we formulate total latency minimization as a mixed integer nonlinear programming (MINLP) problem by considering desired energy consumption, partial offloading, and resource allocation constraints. We use decomposition approach to solve our problem and propose a novel scheme named Joint Partial Offloading and Resource Allocation (JPORA). With aim to reduce the task execution latency, JPORA iteratively adjusts data segmentation and solves the underlying problem of quality of service (QoS)-aware communication resource allocation to the cellular links, and interference-aware communication resource allocation to D2D links. Extensive evaluation results demonstrate that JPORA achieves the lowest latency as compared to the other baseline schemes, meanwhile limiting the local energy consumption of user devices.**

***Index Terms*—Mobile edge computing, D2D communications, partial offloading, latency, resource allocation.**

I. Introduction

THE Internet of Things (IoT) is going to be evolved by the 5G-enabled tactile internet. The researchers at Nokia Bell Labs outlined critical machine communications with relatively low bandwidth and ultralow latency (such as video streaming, home sensors, assisted driving, and remotely controlled vehicles) as one of the 5G technological goals [1]. In this context, the emerging consumer-oriented services demand for the requirements of real-time communication and computation. However, the limited resources of existing devices pose a great challenge to realize the millisecond-scale latency in the future networks [2], [3].

Recently, mobile edge computing (MEC) also known as fog computing has emerged as a promising paradigm to assist various 5G service scenarios including low latency services [4], [5]. Different from the mobile cloud computing architecture, MEC provides powerful computing capabilities at the network edge in close proximity to the end user. Numerous research works proposed mobile edge computation offloading (MECO) for single-user [6]–[9] and multi-user [10]–[13] edge computing scenarios to demonstrated the efficacy of MECO in improving the energy efficiency, delay and utility of task execution.

Keeping in view the resource contention of stand-alone MEC server, the diversity among neighboring devices can be exploited as most mobile users are using less than one-third of their CPU capabilities [14]. In this context, device-to-device (D2D) communication is considered viable as LTE and WiFi-direct technologies enable mobile devices to have multiple network connections simultaneously [15]. Moreover, this technique is particularly attractive for the emerging 5G networks [16] due to the proximity, low latency, better coverage, and traffic offloading gains [17]–[19]. In this respect, many recent works integrated MEC and D2D computation offloading architectures to increase the computation capacity of the MEC network and improve the performance of computation offloading [20]–[25].

It is important to note that, most of the existing works assumed the dedicated spectrum for D2D communications while investigating joint MEC and D2D offloading scenarios. However, some fundamental challenges need to be addressed while minimizing computation offloading latency in a shared spectrum. Specifically, spectrum reuse results in cross-tier interference which increases the transmission latency [26]. Without interference-aware communication resource allocation, the computation offloading latency deteriorates, and meanwhile the power of offloading devices is wasted [27]. Furthermore, resource allocation becomes more challenging when partial offloading model is adopted to benefit from parallel execution [28]. In particular, local energy constraint and distributed capacities in the network result in dynamic data segmentation. Keeping that in view, the

Manuscript received July 13, 2019; revised November 26, 2019 and January 28, 2020; accepted January 28, 2020. Date of publication March 3, 2020; date of current version April 16, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0112300, in part by the National Nature Science Foundation of China under Grants 61861136003, 61621091, and 61673237, in part by the Beijing National Research Center for Information Science and Technology under Grant 20031887521, and in part by research fund of Tsinghua University-Tencent Joint Laboratory for Internet Innovation Technology. The review of this article was coordinated by Prof. L. Zhu. *(Corresponding author: Yong Li.)*

Umber Saleem, Yu Liu, and Yong Li are with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: anb17@mails.tsinghua.edu.cn; liuyu2419@126.com; liyong07@tsinghua.edu.cn).

Sobia Jangsher is with the Department of Electrical Engineering, Institute of Space Technology (IST), Islamabad 44000, Pakistan (e-mail: sobia.jangsher@ist.edu.pk).

Xiaoming Tao is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: taoxm@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TVT.2020.2978027

offloading ratio and interference management need to be jointly considered while allocating resources to minimize the latency in the corresponding scenario.

In this paper, we investigate joint MEC and D2D partial computation offloading to minimize the task execution latency in shared spectrum. In particular, each user can flexibly choose to partially offload its task either to the smart base station (SBS) equipped with MEC server or a nearby device via cellular or D2D link, respectively. Furthermore, we assume that D2D links reuse the cellular communication resources based on orthogonal frequency-division multiple access (OFDMA) scheme. The fundamental challenge here is to strike the right balance between local execution and offloading latencies in presence of dynamic data segmentation, distributed computing capacities, and interference between cellular and D2D transmissions. Therefore, we jointly consider these aspects to formulate a mixed integer nonlinear programming (MINLP) problem of sum latency minimization. We propose a low complexity heuristic which solves the formulated problem by decomposing it into data segmentation and resource allocation problems for known user association.

A. Contribution and Organization

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

1) We propose a D2D-enabled MEC framework that facilitates partial computation offloading to minimize the latency in shared spectrum. Keeping in view the dynamic data segmentation and cross-tier interference, we provide an integrated framework for partial computation offloading and interference management. Accordingly, we jointly formulate the aspects of user association, partial offloading, and resource allocation to minimize the total latency by taking account of local energy consumption and task completion deadline.
2) We formulate latency minimization into MINLP problem and devise a novel scheme named Joint Partial Offloading and Resource Allocation (JPORA) by decomposing the original problem. JPORA iteratively adjusts data segmentation under the execution delay and desired energy consumption bounds, and meanwhile performs quality of service (QoS) and interference-aware communication resource allocation for the cellular and D2D links, respectively.
3) We perform extensive evaluations under various system parameters to validate the performance of our proposed scheme. Evaluation results show that for maximum number of user devices, JPORA scheme achieves approximately 46% less average latency than the delay threshold. Moreover, the comparison shows that JPORA reduces the total latency by 67%, 40%, 50%, and 60% than local computing, random offloading, complete offloading, and Lyapunov method based partial offloading and resource allocation (LPORA) [21] schemes, respectively.

The rest of the paper is organized as follows. Section II discusses the relevant research works. Section III introduces the system model including the details on device, task and partial offloading models. Section IV discusses the formulation of total latency minimization problem, while Section V describes our problem decomposition methodology. Section VI provides the details on our proposed algorithm JPORA and describes the underlying algorithms in detail. Section VII provides performance analysis, followed by the conclusion in Section IX.

II. Related Work

Since computation offloading can release the computation burden of devices, MECO has been investigated in various scenarios such as Fi-Wi, ultra-dense, vehicular edge computing, and UAV-aided edge computing networks to reduce energy or latency of task execution [29]–[31]. To this end, joint communication and computation resource management has received much attention in recent years. The respective works can be categorized based on the user scenario as single-user [6]–[9], multi-user [10]–[13], and multi-user cooperative edge computing [20]–[25]. Meanwhile, two computation task models have been adopted in these works, namely, binary offloading and partial offloading. The former is adopted when task is atomic and has to be executed as a whole either locally or offloaded to the edge server. The latter deals with the tasks composed of multiple components, and hence can be partitioned for local and remote executions.

For the simple case of single-user systems comprising of a mobile device and MEC server, the selection of execution mode (either local or remote) was mainly investigated while considering binary offloading [6], [7]. However, partial offloading is more suitable for the applications with stringent latency requirements, since it benefits from parallelism between local and remote executions. With this regard, several works adopted partial offloading and investigated the optimal offloading ratio to further enhance the computation offloading performance [8], [9]. These works demonstrated that partial offloading can achieve lower latency and higher energy savings as compared to binary offloading.

Different from the preceding works, multi-user MEC system was considered in [10]–[13]. As multiple users compete for finite radio and edge computing resources, the main challenge is to design joint radio and computation resource allocation policies. In particular, the authors adopted partial offloading model and jointly investigated data segmentation and resource allocation to minimize latency [10]–[12] and energy [13] of task offloading. It is important to note that, the data segmentation strategies derived in [10] and [11] didn't consider the local energy constraint of mobile user while minimizing the latency by adopting partial offloading model. However, Saleem *et al.* [12] formulated latency minimization problem by taking account of local energy constraint while keeping in view that limited energy availability at user impacts the data segmentation decision. Despite the manifold benefits, some shortcomings are inevitable. Aggressive offloading can aggravate service delay due to finite capacity of MEC server and scarce bandwidth. Furthermore, the users with poor channel conditions cannot equally benefit from the stand-alone MEC server. The possible solutions such as deploying new cloudlets or reconfiguring the existing ones are undesirable due to sheer cost of physical servers [32] or demand for optimal load balancing schemes [33], [34], respectively.

In this context, multiuser cooperative edge computing is envisioned as a promising technique, where collaboration among proximal devices can scale up the service without any additional cost. Recently, several works jointly proposed MEC and D2D computation offloading [20]–[25] due to the proximity and multiplexing gains of D2D communication [17]. In particular, Chen *et al.* [20] devised a three-layer graph matching based binary task offloading by transforming minimum total task execution cost problem into a minimum weight matching problem and solving it using Edmonds' Blossom algorithm. Instead of binary offloading, the authors in [21]–[23] adopted partial offloading to benefit from the parallelism in a D2D-enabled MEC scenario. Assuming that task can be simultaneously executed locally and offloaded to MEC server and proximal device, He *et al.* [21] formulated MINLP problem of computation capacity maximization. Using decomposition approach, the authors first minimized the required edge computation resource for a given D2D pair by optimal power allocation, and then maximized the number of supported devices by optimal D2D pairing. On the other hand, Ti *et al.* [22] minimized the weighted energy consumption of users, helper nodes, and edge cloud by jointly optimizing the computation offloading and resource allocation with latency, radio, and computational resource constraints. By considering the problem to be signomial geometric programming, the authors adopted successive convex approximation method which iteratively solves a sequence of approximated convex problems. The authors in [23] addressed the problem of computation offloading, mode selection, and the resource allocation strategy design for cellular D2D MEC system. A heuristic algorithm was proposed, which successively solved computation offloading and resource allocation subproblems by KuhnMunkres algorithm and Lagrangian dual method, respectively.

It is important to note that, the works in [20]–[23] overlooked the transmission interference between cellular and D2D links by considering dedicated spectrum. In general, D2D communication operating in underlay mode shares the cellular spectrum by frequency reuse. Although spectrum reuse enhances spectral efficiency, the resulting interference demands for optimal spectrum allocation to improve the computation offloading performance in such scenario. Hu *et al.* [24] formulated the multi-user multi-destination binary offloading decision using sequential game by taking into account the interference to minimize the computation offloading overhead. Similarly, Wang *et al.* [25] also considered the binary offloading decision along with wireless resource selection to minimize the energy consumption. The authors introduced the potential game approach by approximating the interference.

Although the prior works in [20]–[25] have been dedicated to studying D2D-enabled MEC system, the problem of partial offloading latency minimization while preserving user energy in a shared spectrum has not been addressed so far. In this regard, we highlight several gaps in the relevant literature that had been unexplored. First, while considering partial offloading in D2D-enabled MEC system, the spectrum sharing needs to be explored. Specifically, the interference-aware resource allocation needs to be investigated while taking into account the dynamic data segmentation. Second, local energy constraint plays a critical role in partial offloading decision which has been ignored by the similar problems in [20]–[23]. Third, the aspects of user association, partial offloading and resource allocation to reduce the computation offloading delay with spectrum reuse are generally considered separately in the prior works. However, it is necessary to derive optimal task scheduling policies by jointly addressing these aspects when considering flexible offloading in D2D-enabled MEC network.

III. System Model

In this section, we describe the system model based on our consideration of joint D2D and MEC offloading. We first present the network model which consists of cellular and D2D links. Then we describe the communication, task, and partial offloading models in detail.

A. Network Model

We consider a network with a SBS and multiple user devices, where users have flexible offloading choice between D2D or SBS offloading. The devices in the network are divided into two types, namely, task devices (TDs) and resource devices (RDs). The set $\mathcal{M} = \{1, 2, \ldots, M\}$ of TDs consists of all such user devices which have limited computation resources and a delay sensitive computation intensive task to perform. On the other hand, RDs include SBS 0, and set of idle user devices with relatively higher computation capability denoted as $\mathcal{K} = \{1, 2, \ldots, K\}$. Hence, the set of RDs is defined as $\mathcal{S} = \{0, \mathcal{K}\}$.

Due to limited computation capability, a TD can partially offload its computation to one of the RDs. We assume that each TD is aware of the location of neighboring D2D RDs and the SBS. Based on its own location, a TD can either offload its computation to the SBS or a D2D RD via a cellular or a D2D link, respectively. A quasi-static network scenario is assumed where location of the users remain unchanged during the computation offloading period [35]. To represent the user association, we introduce a binary indicator variable $x_{ms} \in \{0, 1\}$, where $m \in \mathcal{M}$ and $s \in \mathcal{S}$, and define the user association profile as $\boldsymbol{x} = \{x_{ms} | s \in \mathcal{S}, m \in \mathcal{M}\}$. We further discuss the user association and the respective links in the network as follows

- *Cellular Link:* A TD can partially offload its computation task to the SBS via a wireless cellular link. The indicator variable $x_{m0} = 1$ represents that the task of TD m is offloaded for execution at the SBS 0 through a cellular link. SBS can serve multiple TDs simultaneously. However, the number of served TDs is limited by the finite computation capacity of the SBS. The set of cellular TDs is defined as $\mathcal{U}_c = \{m | x_{m0} = 1, \forall m \in \mathcal{M}\}$.
- *D2D Link:* A TD can establish a direct D2D link with any of the D2D RDs from the set $\mathcal{K} = \{1, 2, \ldots, K\}$ within the maximum distance of $d^{\max}$ (using technologies such as WiFi-Direct or Bluetooth). Here, we assume that the D2D connection and transmission is secured using widely adopted public key cryptography technique [14], [36]. The

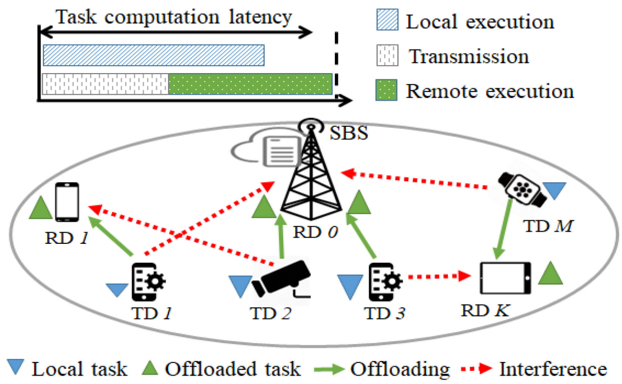


Fig. 1. Illustration of communication and computation resource sharing in D2D-enabled partial computation offloading.

TABLE I
NOTATIONS

Notation	Definition
$\mathcal{M}$	Set of TDs
$\mathcal{K}$	Set of D2D RDs
$\mathcal{S}$	Set of RDs including SBS and D2D RDs
$\mathcal{N}$	Set of subcarriers
m	Index of TD
s	Index of RD
n	Index of subcarrier
D_m	Task data size of TD m
C_m	Task processing density of TD m
τ_m	Delay threshold required by TD m
E_m	Desired local energy consumption of TD m
α_m	Partial offloading fraction of TD m
ρ_{ms}^n	Subcarrier assignment indicator
p_{ms}^n	Power allocation parameter
x_{ms}	User association indicator
f_{ms}	Computation resource allocation parameter

indicator variable $x_{mk} = 1$ shows that the task of TD m is offloaded through a D2D link to RD $k \in \mathcal{K}$ for remote execution, and thus a D2D link is indicated by x_{mk}. For a particular time frame, the feasible D2D links remain same for a TD. Moreover, a TD can offload its computation to at most one D2D RD, and a D2D RD can serve at most one offloading device resulting in non-overlapping D2D pairs in the network. The set of D2D TDs is defined as $\mathcal{U}_d = \{m | x_{mk} = 1, \forall m \in \mathcal{M}, \forall k \in \mathcal{K}\}$.

A simple illustration of considered scenario is shown in Fig. 1, which depicts computation resource sharing by partial offloading, and communication resource sharing among cellular and D2D links resulting in interference. The important notations used in the rest of this paper are summarized in Table I.

B. Task Model

We assume that each TD has a delay sensitive application task with large input data to process. Specifically, we consider the data partitioned oriented applications. For such applications, the input data is known beforehand and can be arbitrarily partitioned for parallel processing due to bit-wise independence. The typical examples are virus scan, file/figure compression, recognition, and vision applications [37]. We consider the computation offloading problem over a short time frame, and thus each user has only one task to offload at a particular time. A task of TD $m \in \mathcal{M}$ can be described as $\mathcal{I}_m \triangleq < D_m, C_m, \tau_m >$, where D_m is the data size of the task (measured in bits), C_m is the computation resource required to compute one bit of the task (measured in CPU cycles per bit), and τ_m is the task deadline i.e. maximum tolerable delay of task execution (in seconds).

C. Partial Offloading Model

Motivated by the fact that partial offloading benefits from parallel computing by efficiently utilizing the local and remote resources simultaneously [29], we adopt partial computation offloading model. In particular, we adopt data-partition model assuming the applications to be data partitioned oriented [37]. It is the simplest partial offloading model, where the input bits of the task can be arbitrarily divided due to bit-wise independence [28]. Therefore, we assume the partition cost to be negligible similar to the relevant works in [10]–[13].

Assuming that a TD's application data can be partitioned, a fraction of task is processed locally, and the rest is offloaded for remote execution on the RD. We introduce the offloading fraction parameter as $\alpha_m \in [0, 1]$. So we have $\boldsymbol{\alpha} = \{\alpha_m | m \in \mathcal{M}\}$ as the offloading fraction profile. After determining the offloading fraction value, $(1 - \alpha_m)D_m$ bits are processed locally, while $\alpha_m D_m$ bits are offloaded for execution at a remote device. Therefore, the computation of a task involves local execution, communication and remote execution delays, which are discussed in the following sections in detail.

1) Local Execution: Each TD has a fixed CPU frequency, while it may vary over different devices. For a TD $m \in \mathcal{M}$, let f_m denote the computation capacity measured in CPU cycles per second. A TD will process a fraction of its task locally, whereas the time consumption of local computation depends on the CPU clock frequency f_m and the number of CPU cycles required per bit C_m. Then the local computation latency L_m^{loc} at TD m is given as

$$L_m^{\text{loc}} = \frac{(1 - \alpha_m) D_m C_m}{f_m}. \tag{1}$$

Let $\mathcal{W}_m$ denote the energy consumption per CPU cycle for local computing at TD m. Then $C_m \mathcal{W}_m$ gives the computing energy per bit. The energy consumed for local computing is given as

$$E_m^{\text{loc}} = (1 - \alpha_m) D_m C_m \mathcal{W}_m. \tag{2}$$

2) Communication: We use OFDMA scheme in the uplink, and thus the available bandwidth B is divided into N orthogonal subcarriers. Let $\mathcal{N} = \{1, 2, \ldots N\}$ denote the subcarrier set, where $n \in \mathcal{N}$ represents a particular subcarrier. We introduce binary subcarrier assignment parameter $\rho_{ms}^n \in \{0, 1\}$ and the subcarrier assignment profile $\boldsymbol{\rho} = \{\rho_{ms}^n | m \in \mathcal{M}, s \in \mathcal{S}, n \in \mathcal{N}\}$. Whereas, $\rho_{ms}^n = 1$ indicates that a TD $m \in \mathcal{M}$ linked to RD $s \in \mathcal{S}$ is assigned the subcarrier $n \in \mathcal{N}$, and verse vice. For efficient spectrum utilization, we assume frequency reuse such that a subcarrier can be shared by at most two devices simultaneously.

Let h_{ms}^n denote the channel gain between TD $m \in \mathcal{M}$ and RD $s \in \mathcal{S}$ on subcarrier $n \in \mathcal{N}$ corresponding to a white Gaussian noise channel, which incorporates distance based path loss model and independent Rayleigh fading. Let p_{ms}^n denote the transmission power of link from TD m to RD s on subcarrier n. The power allocation profile is defined as $\boldsymbol{p} = \{p_{ms}^n | m \in \mathcal{M}, s \in \mathcal{S}, n \in \mathcal{N}\}$. Then the data rate r_{ms}^n from TD m to RD s on subcarrier n can be expressed using Shannon formula as

$$r_{ms}^n = \bar{B}\log_2(1 + \gamma_{ms}^n), \tag{3}$$

where $\bar{B}$ is the bandwidth per subcarrier and γ_{ms}^n is the signal to interference plus noise ratio (SINR). In case of no reuse, $\gamma_{ms}^n = x_{ms}\rho_{ms}^n p_{ms}^n h_{ms}^n / N_0$, where N_0 is the noise power per subcarrier and there is no interference. In case a D2D link reuses the spectrum of a cellular link, the SINR of cellular link can be expressed as

$$\gamma_{m0}^n = \frac{x_{m0}\rho_{m0}^n p_{m0}^n h_{m0}^n}{N_0 + \sum_{m'=1, m' \neq m}^{M} \sum_{k=1}^{K} x_{m'k}\rho_{m'k}^n p_{m'k}^n h_{m'0}^n}, \tag{4}$$

where $h_{m'0}^n$ is the channel gain of interference link between D2D TD and the SBS.

The reuse of spectrum is allowed only to a cellular and a D2D link, and not among the D2D links. Therefore, the SINR of D2D link when it reuses subcarrier of a cellular link can be expressed as

$$\gamma_{mk}^n = \frac{x_{mk}\rho_{mk}^n p_{mk}^n h_{mk}^n}{N_0 + \sum_{m'=1, m' \neq m}^{M} x_{m'0}\rho_{m'0}^n p_{m'0}^n h_{m'k}^n}, \tag{5}$$

where $h_{m'k}^n$ is the channel gain of interference link between the cellular TD and the D2D RD.

The total achievable data rate of TD m can be written as

$$R_m = \sum_{s=0}^{K} \sum_{n=1}^{N} r_{ms}^n. \tag{6}$$

The total transmission power of a TD m in uplink is $P_m = \sum_{s=0}^{K} \sum_{n=1}^{N} x_{ms}\rho_{ms}^n p_{ms}^n$, while the maximum power allocated to a TD in uplink is limited by $P_m^{\max}$.

In order to guarantee the reduction in communication cost, the QoS constraint is imposed on each TD. Hence, we assume that the data rate of a TD should be greater than a minimum threshold of $R_m^{\min}$. As a consequence, the number of subcarriers assigned to a TD should satisfy its QoS at least with equality.

For computation offloading, the input data is transmitted through the wireless channel resulting in communication delay. For simplicity, we assume that the offloading data is uniformly distributed over all the subcarriers of TD. Let N_m denote the total number of subcarriers assigned to a TD m and its value is obtained as $N_m = \sum_{s=0}^{K} \sum_{n=1}^{N} x_{ms}\rho_{ms}^n$. Then the data per subcarrier is $\bar{d}_m = \alpha_m D_m / N_m$. Due to multi-channel transmission, the offloading latency L_m^{off} of a TD m is determined by the transmission delay of its worst subcarrier and can be expressed as

$$L_m^{\text{off}} = \max_{n \in \mathcal{N}} \left(\frac{\bar{d}_m}{\sum_{s=0}^{K} x_{ms} r_{ms}^n} \right). \tag{7}$$

Whereas, the total energy consumption for offloading a task of TD m can be expressed in terms of task size, transmission power, and transmission rate as

$$E_m^{\text{off}} = \sum_{s=0}^{K} \sum_{n=1}^{N} \frac{x_{ms}\rho_{ms}^n p_{ms}^n \bar{d}_m}{r_{ms}^n}. \tag{8}$$

3) Remote Execution: Once a task is offloaded to the SBS or a D2D RD, it is executed remotely, and the results are downloaded at the TD. Let f_{ms} denote the computation resource in CPU cycles per second allocated to a TD $m \in \mathcal{M}$ at RD $s \in \mathcal{S}$ for executing the offloaded task. The computation resource allocation profile is defined as $\boldsymbol{f} = \{f_{ms} | m \in \mathcal{M}, s \in \mathcal{S}\}$. The remote execution latency for TD m is given as

$$L_m^{\text{rem}} = \sum_{s=0}^{K} \frac{x_{ms}\alpha_m D_m C_m}{f_{ms}}. \tag{9}$$

As we assume finite computation capacity at the SBS 0, a feasible computation resource allocation at the SBS must satisfy $\sum_{m=1}^{M} x_{m0} f_{m0} \leq F_0$, where F_0 is the total computation capacity of the SBS in CPU cycles per second. On the other hand, the computation resources of D2D RD are completely allocated to the offloaded task as only one TD is served by a D2D RD.

Hitherto we described the D2D-enabled MEC scenario and discussed the partial offloading model consisting of three stages: local execution, communication, and remote execution. We proceed to formulate the joint user association, data segmentation, and resource allocation problem in next section to minimize the users latency with QoS guaranteed.

IV. Problem Formulation

In this work, we aim to minimize the sum of task execution latency for all the TDs which partially offload their computation either to the SBS or a nearby RD by establishing a cellular or a D2D link. Once the offloading decision is made at a TD, the data fraction for remote execution is transmitted over the communication links to the associated RD. The RD executes the task when transmission is completed. In case of partial offloading, there are two processes involved, namely, local computation, and computation offloading (offloading plus remote execution). Since local computation can undergo simultaneously with the computation offloading process, the total task computation delay for a user m is determined by the longer process and can be expressed as

$$L_m = \max\{L_m^{\text{loc}}, (L_m^{\text{off}} + L_m^{\text{rem}})\}. \tag{10}$$

Based on the system model in Section III, we formulate a joint partial offloading, communication and computation resource allocation problem as follows:

$$\textbf{P1:} \quad \min_{\boldsymbol{x}, \boldsymbol{\alpha}, \mathbf{p}, \boldsymbol{\rho}, \mathbf{f}} \sum_{m=1}^{M} L_m \tag{11a}$$

$$\text{s.t.} \quad L_m \leq \tau_m, \ \forall m \in \mathcal{M}, \tag{11b}$$

$$E_m^{\text{loc}} + E_m^{\text{off}} \leq E_m, \ \forall m \in \mathcal{M}, \tag{11c}$$

$$\sum_{s=0}^{K} x_{ms} \leq 1,\ \sum_{m=1}^{M} x_{mk} \leq 1,\ \forall k \in \mathcal{K}, \tag{11d}$$

$$x_{ms} \in \{0,1\},\ \forall s \in \mathcal{S}, \forall m \in \mathcal{M}, \tag{11e}$$

$$\sum_{m=1}^{M}\sum_{n=1}^{N} x_{m0}\rho_{m0}^{n} \leq N, \sum_{s=1}^{K}\sum_{m=1}^{M}\sum_{n=1}^{N} x_{ms}\rho_{ms}^{n} \leq N, \tag{11f}$$

$$\sum_{s=0}^{K}\sum_{m=1}^{M} x_{ms}\rho_{ms}^{n} \leq 2,\ \forall n \in \mathcal{N}, \tag{11g}$$

$$\sum_{k=1}^{K}\sum_{m=1}^{M} x_{mk}\rho_{mk}^{n} \leq 1,\ \sum_{m=1}^{M} x_{m0}\rho_{m0}^{n} \leq 1,\ \forall n \in \mathcal{N}, \tag{11h}$$

$$\rho_{ms}^{n} \in \{0,1\},\ \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{11i}$$

$$0 \leq \sum_{s=0}^{K}\sum_{n=1}^{N} x_{ms}\rho_{ms}^{n}p_{ms}^{n} \leq P_{m}^{\max},\ \forall m \in \mathcal{M}, \tag{11j}$$

$$\sum_{s=0}^{K}\sum_{n=1}^{N} x_{ms}r_{ms}^{n} \geq R_{m}^{\min},\ \forall m \in \mathcal{M}, \tag{11k}$$

$$\sum_{m=1}^{M} x_{m0}f_{m0} \leq F_0, \tag{11l}$$

$$0 \leq \alpha_m \leq 1,\ \forall m \in \mathcal{M}. \tag{11m}$$

Here, the objective function in (11a) aims to minimize the total latency which is the sum of task execution latency of all the TDs. The constraints in (11b) imply that the task execution latency of TD m must not exceed the maximum tolerable delay. Whereas (11c) shows the energy efficiency constraints, which ensure that the energy consumption of computation offloading process $(E_m^{\text{loc}} + E_m^{\text{off}})$ for TD m must not exceed the expected energy cost E_m. The user association constraints are presented in (11d) and (11e). In (11d), the first part shows that TD m is either linked to the SBS via a cellular link or to the D2D RD via a D2D link, while the second part shows the one-to-one D2D association constraints. The constraints in (11e) show that user association is a binary variable. The communication resource allocation constraints are presented in (11f)-(11k). Specifically, (11f) shows that the total subcarriers assigned to all the cellular and the D2D links are limited by the number of available subcarriers. The constraints in (11g) and (11h) present the subcarrier reuse conditions, where (11g) means that subcarrier n can be shared by at most two links, and (11h) implies that only a cellular and a D2D link are allowed to reuse subcarrier n. The constraints in (11i) show that subcarrier assignment is a binary variable. The constraints in (11j) and (11k) depict the maximum power and the QoS constraints of TD m in uplink, respectively. The constraint in (11l) present feasible computation resource allocation at the SBS, which means that the computation resources are allocated to the offloading TDs within the computation capacity of the SBS. Finally, the constraints in (11m) bound the offloading fraction value of TD m.

It is noticeable that, the total latency minimization problem formulated above is MINLP problem. The offloading fraction $\boldsymbol{\alpha}$, power allocation $\boldsymbol{p}$, and computation resource allocation $\boldsymbol{f}$ have continuous values, while user association $\boldsymbol{x}$ and subcarrier assignment $\boldsymbol{\rho}$ have binary values. The objective function in (11a) is non-convex due to the mixed nature of optimization variables. Furthermore, the product relation among integer and continuous variables results in nonlinear constraints and non-convex solution set. In terms of complexity, the formulated problem is NP-hard, and we cannot obtain its solution in polynomial time [38].

Theorem 1: The total latency minimization problem in **P1** is NP-hard.

Proof: To prove the NP-hardness of problem **P1**, we first consider a special case of the problem where the user association is known. Also, the data segmentation decision is fixed such that the local energy consumption constraint is not violated. As a consequence, the problem **P1** is rendered to latency minimization by optimal communication resource allocation. This special case then can be easily reduced to multiple knapsack problem, which is a well-known NP-hard problem [38]. ■

Multiple knapsack problem [39]: There is a set of items and set of knapsacks, where each item has a certain profit and weight, while each knapsack has certain capacity. The problem of multiple knapsack is to select and assign disjoint subsets of items to a unique knapsack, such that the total profit is maximized while the capacity of knapsack is at least the total weight of the selected items. In this context, we can regard the set of M TDs and N wireless channels as the items and knapsacks, respectively. The amount of resources required by each TD is the weight, while the computation offloading latency is the profit. Then, filling the items to knapsacks is equivalent to assigning the TDs to the wireless channels such that the required resources are satisfied, and meanwhile the latency is reduced. As this special case of problem **P1** can be mapped to multiple knapsack, it can be inferred that problem **P1** is also NP-hard.

V. Problem Decomposition

One approach to solve problem **P1** is relaxation and rounding, where the binary variable is relaxed and the integer constraints in (11i) are changed to real-value constraints as $0 \leq \rho_{ms}^{n} \leq 1$. The resulting convex problem is then solved using any of the convex optimization techniques, and the integer solution is obtained using rounding techniques [40]. Such an approach is infeasible for our problem as the relaxed problem would still have non-convex objective function due to the data rate term in the denominator with interference expression.

It can be noted from the proposed optimization problem that for known user association, the constraints in (11b) and (11c) can be decoupled from the communication and computation resource allocation constraints in (11f)(11l). Therefore, we solve our problem by the decomposition method. For known user association, we first decide the data segmentation policy based on the latency and energy constraints in (11b) and (11c). Then, the original problem is transformed into resource allocation problem. We propose an iterative algorithm which adjusts the value of offloading fraction and solves the underlying interference-aware resource allocation problem to improve the average latency of the network.

We decide the user association based on the location of each TD. Keeping in view that the users close to the cell boundary mostly suffer from higher offloading delay due to bad channel conditions, we allow D2D offloading to the distant TDs. We define a constant λ as a threshold value of distance from the SBS. We determine many-to-one cellular user associations for all the TDs located within the range λ. On the other hand, the TDs located beyond the range λ can only perform D2D offloading according to our assumption. Therefore, one-to-one user association is determined for these TDs based on maximum distance of $d^{\max}$ between the TD and the D2D RD. In this way, the user association $\boldsymbol{x}$ is decided, and the corresponding cellular and D2D link sets are obtained.

A. Data Segmentation Policy

An important factor for partial offloading is to determine the optimal fraction of data offloaded by a user, as it effects both the time consumption for local execution, offloading and remote execution, and the energy consumption for local computing and offloading. Based on (11b) and (11c), a delay bounded and energy efficient data segmentation policy can be derived. Therefore, our proposed data segmentation strategy is influenced by two assumptions. Firstly, the offloading fraction should be decided such that the two parallel processes are completed within the delay threshold of a TD. Secondly, the energy cost of offloading should not exceed the desired energy consumption of a TD.

It is obvious that the minimum latency for a TD m is reached when the two parallel processes (local execution and offloading plus remote execution) take the same time. Then under the constraint (11b) on the total latency, the L_m^{loc} is bounded as

$$\frac{(1-\alpha_m)D_m C_m}{f_m} \leq \tau_m. \tag{12}$$

The constraint in (12) provides feasibility condition in terms of a lower bound on α_m, which can be easily obtained as

$$\alpha_m^{\min} = 1 - \frac{\tau_m f_m}{D_m C_m}. \tag{13}$$

Under the constraint (11c) on offloading energy consumption, an upper bound on α_m is obtained as

$$\alpha_m^{\max} = \frac{D_m C_m \mathcal{W}_m - E_m}{D_m \left(C_m \mathcal{W}_m - \sum_{s=0}^{K}\sum_{n=1}^{N} x_{ms}\rho_{ms}^n p_{ms}^n / r_{ms}^n\right)}, \tag{14}$$

where $D_m C_m \mathcal{W}_m$ gives the energy consumption when the complete task is executed locally. As the communication resource allocation is not known here, initially we use $E_m^{\text{off}} = \alpha_m D_m P_m / R_m^{\min}$ to derive the value of upper bound on α_m and the expression in (14) is modified as $D_m C_m \mathcal{W}_m - E_m / (D_m C_m \mathcal{W}_m - D_m P_m / R_m^{\min})$. It is important to note that we are assuming the minimum data rate $R_m^{\min}$ here which results in maximum energy consumption. Therefore, the use of aforementioned expression is justified as it corresponds to the worst case of energy consumption.

B. Communication and Computation Resource Allocation

Based on the known user association $\boldsymbol{x}$ and the data segmentation policy, we can transform the original problem **P1** to latency minimization by communication and computation resource allocation problem as follows:

$$\textbf{P2:} \quad \min_{\mathbf{p},\boldsymbol{\rho},\boldsymbol{f}} \sum_{m=1}^{M} L_m \tag{15a}$$

$$\text{s.t.} \quad L_m^{\text{off}} + L_m^{\text{rem}} \leq \tau_m, \ \forall m \in \mathcal{M}, \quad \text{(11f)–(11l)}, \tag{15b}$$

where α_m is set to $\alpha_m^{\min}$ initially, which bounds the local computation time within the delay threshold. Thus, (11b) from original problem is modified here to emphasize the latency constraint for offloading process specifically.

Problem **P2** is still non-convex due to the product of integer and real valued variables. We simplify the computation resource allocation constraint for remote execution. For the TDs offloading to the SBS, we decide uniform resource allocation and obtain the computation resource allocated to a TD m at SBS 0 as $f_{m0} = F_0 / |\mathcal{U}_c|$. Now we have to solve the latency minimization problem by optimal communication resource allocation in presence of intra-cell interference, which is discussed in later sections.

C. Data Rate Estimation

Since the data segmentation policy is known, the corresponding value of minimum data rate required to offload $\alpha_m D_m$ bits can be obtained using $\alpha_m D_m / L_m^{\text{off}}$. Under the latency constraint in (11b), $L_m^{\text{off}} + L_m^{\text{rem}} \leq \tau_m$ and thus $R_m^{\min}$ can be obtained as

$$R_m^{\min} = \frac{\alpha_m D_m}{|\tau_m - L_m^{\text{rem}}|}. \tag{16}$$

We use absolute value in the denominator as the edge execution time may exceed the delay threshold initially. However, our algorithm iteratively improves the offloading and remote execution latencies to satisfy the delay constraint.

VI. Joint Partial Offloading and Resource Allocation (JPORA) Scheme

We propose an iterative algorithm, which adjusts the data segmentation and solves the underlying problem of QoS-aware resource allocation for cellular links followed by interference-aware resource allocation for D2D links. We first discuss these underlying algorithms in detail and then describe the global algorithm for JPORA and discuss its complexity.

A. Communication Resource Allocation

In the considered scenario of D2D assisted computation offloading in cellular network, the cross tier interference (between

Algorithm 1: QoS Aware Communication Resource Allocation for Cellular TDs.

1: **Input:** $\mathcal{N}, \mathcal{U}_c, \mathcal{U}_d, \boldsymbol{x}, R_m^{\min}$, Channel gains $\forall m \in \mathcal{M}$.
2: **Output:** $\boldsymbol{\rho}, \boldsymbol{p}$.
3: **Initialize:** $U = \mathcal{U}_c, U' = \mathcal{U}_c, S = \mathcal{N}, N_m = 0, \boldsymbol{R}_m = 0, \forall m \in \mathcal{M}$.
4: **while** $U \neq \phi$ **do**
5: Find $(m^*, n^*) = \arg\max_{n \in S, m \in U}(\frac{r_{m0}^n}{\sum_{m' \in U'} r_{m'0}^n / |\mathcal{U}_c|})$.
6: Set $\boldsymbol{\rho}(m^*, n^*) = 1$ and update N_{m^*}.
7: Set $\boldsymbol{p}(m^*, n^*) = P_m^{\max}$ and $\boldsymbol{R}_m(m^*) = r_{m^*0}^{n^*}$.
8: $U = U - \{m^*\}$.
9: $S = S - \{n^*\}$.
10: **end while**
11: **while** $S \neq \phi$ **do**
12: Find $m' = \arg\min_{m \in U}(\frac{R_m}{R_m^{\min}})$.
13: Find $n^* = \arg\max_{n \in S}(\frac{r_{m'}^n}{\frac{1}{M}\sum_{m \in U'} r_{m'}^n})$.
14: Set $\boldsymbol{\rho}(m', n^*) = 1$ and update $N_{m'}$.
15: Update power allocation $\boldsymbol{p}$ for m' using $P_{m'}^{\max}/N_{m'}$.
16: Update $\boldsymbol{R}_m(m') = \boldsymbol{R}_m(m') + r_{m'0}^{n^*}$ using $\boldsymbol{p}$.
17: $S = S - \{n^*\}$.
18: **end while**

cellular and D2D links) arises due to the spectrum reuse [41]. While reuse of spectrum improves spectral efficiency, the QoS of primary users is degraded. As a consequence, the offloading performance is degraded. This demands for interference-aware communication resource allocation. In order to improve the latency, we discuss our communication resource allocation scheme which improves the latency performance by mitigating the interference and ensures spectral efficiency at the same time.

We perform communication resource allocation based on interference mitigation in Algorithm 1 and Algorithm 2. We discuss the steps in these algorithms in detail as follows.

1) QoS-Aware Subcarrier Assignment and Power Allocation for Cellular TDs: Algorithm 1 performs QoS-aware communication resource allocation for all the cellular TDs $(m \in \mathcal{U}_c)$ considering no reuse. A subcarrier n should be assigned to a TD m such that $m = \arg\max_i(r_i^n)$. Assuming varying channel gain across all the subcarriers, we first assign one subcarrier to each cellular TD based on maximum marginal data rate in Line 5 [12]. As there is only one subcarrier per cellular link initially, maximum power is allocated to each cellular TD in uplink. We then assign the remaining subcarriers such that each time we determine a TD whose QoS is least met (by finding the smallest value of $R_m/R_m^{\min}$ using (6) and (16)). Employing the marginal data rate approach, we determine an optimal subcarrier for the weakest cellular link (in terms of sum rate) in Line 13.

Keeping in view that each subcarrier holds equal amount of data, we decide uniform power allocation in uplink. Therefore, the maximum power $P_m^{\max}$ is distributed uniformly among all the assigned subcarriers of TD m. Each time a subcarrier is assigned to a TD, its power allocation is also updated accordingly and corresponding sum rate is computed. The subsequent iterations improve the sum rate of all the cellular links beyond the QoS requirement with an intent to reduce the transmission delay per subcarrier.

Next, we discuss the computational complexity of Algorithm 1. The first while loop has $|\mathcal{U}_c|$ iterations for initial subcarrier assignment to the cellular TDs. The complexity of searching an optimal TD and subcarrier pair is $O(N|\mathcal{U}_c|)$ in each iteration. Thus the complexity of initial subcarrier assignment is $O(N|\mathcal{U}_c|^2)$. For assigning the remaining subcarriers in second while loop, the complexity is $O(|\mathcal{U}_c|(N - |\mathcal{U}_c|)^2)$. Since $|\mathcal{U}_c| \leq M$ and in practice $N >> M$, the worst case complexity of Algorithm 1 can be represented as $O(N^2 M)$.

2) Interference Aware Communication Resource Allocation for D2D TDs: Once the communication resources are assigned to the cellular links, we determine optimal subcarrier assignment and power allocation to the D2D links based on minimum interference in Algorithm 2. There are two main steps in Algorithm 2, which are discussed here in detail.

In the first step, we determine D2D TD to share the communication resource of a particular cellular TD. To avoid degradation of weak cellular links, we first arrange all the cellular TDs in descending order of their data rates. We then sequentially select a cellular TD with higher data rate, determine its reuse partner, and perform resource allocation. The process is repeated until all the D2D TDs are assigned communication resources. Assuming that the gain factor provides a realistic measurement of the channel, we select a reuse partner for a cellular TD using the classical approach of channel gain. For a cellular TD c, we find a D2D TD d with the minimum average gain of the interference link between c and d's RD k denoted as $\bar{h}_{ck}$. It is important to note that while determining a reuse pair we focus on the interference at the D2D RD as a cellular TD is allocated maximum power in uplink. Moreover, owing to the large distance from the SBS, the interference created by D2D TD at the SBS becomes trivial.

After deciding an optimal reuse pair (c, d), the D2D TD d is assigned all the subcarriers of the cellular TD c. Next, we decide the power allocation for d such that the QoS constraint of both c and d is satisfied. As the power of cellular TD is fixed to maximum value $P_c^{\max}$, the required power for D2D TD P_d^{req} lies in the interval $[P_d^l, P_d^r]$. Thus, the optimal power allocation for a reuse pair resides on one of the following points $\{(P_c^{\max}, P_d^l), (P_c^{\max}, P_d^r)\}$. We derive the values of P_d^l in Line 9 and P_d^r in Line 15, based on the minimum SINR requirement of D2D $(\gamma_d^{\min})$ and cellular TD $(\gamma_c^{\min})$, respectively. Whereas, the minimum SINR corresponds to minimum data rate obtained using (16). Keeping in view the close proximity of D2D links, the minimum power is decided as the required power for D2D TD. Similar to the power allocation for cellular links, P_d^{req} is uniformly distributed among all the subcarriers of TD d. Finally, the cellular and D2D TDs in reuse pair are removed from the corresponding sets.

Next, we discuss the complexity of Algorithm 2. There are $\mathcal{U}_d$ iterations of for loop, while in each iteration the complexity of searching an optimal reuse partner for a particular cellular link is $O(|\mathcal{U}_d|^2)$. Since $|\mathcal{U}_d| \leq M$, the worst case complexity of Algorithm 2 is $O(M^2)$.

Algorithm 2: Interference Aware Communication Resource Allocation for D2D TDs.

Input: $\mathcal{N}, \mathcal{U}_c, \mathcal{U}_d, \boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{p}, R_m^{\min}$, Channel gains $\forall m \in \mathcal{M}$.
Output: $\boldsymbol{\rho}, \boldsymbol{p}$.
Initialize: $U = \mathcal{U}_d, U' = \mathcal{U}_c$.
Arrange TDs in U' in descending order of $\boldsymbol{R}_m$.
for $i = 1$ to $|U|$ **do**
  $c = U'(i)$.
  Find reuse pair (c, d) with minimum $\bar{h}_{ck}$.
  Assign $\boldsymbol{\rho}(d, \mathcal{N}) = \boldsymbol{\rho}(c, \mathcal{N})$ and update N_d.
  Set $P_d = \frac{\gamma_d^{\min}(N_0 + P_c^{\max}\bar{h}_{ck})}{\bar{h}_{dk}}$.
  if $P_d < P_c^{\max}$ **then**
    $P_d^l = P_d$.
  else
    $P_c = P_c^{\max}$ is infeasible, return.
  end if
  $P_d^r = \frac{\bar{h}_{c0}P_c^{\max} - \gamma_c^{\min}N_0}{\gamma_c^{\min}\bar{h}_{d0}}$.
  $P_d^{\text{req}} = \min\{P_d^l, P_d^r\}$.
  Update power allocation $\boldsymbol{p}$ for d using P_d^{req}/N_d.
  $U = U - \{d\}$.
  $U' = U' - \{c\}$.
end for

Algorithm 3: JPORA.

1: **Input:** $\mathcal{M}, \mathcal{N}, \mathcal{I}_m\ \forall m, \lambda$, Channel gains $\forall m, s, n$.
2: **Output:** $\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\rho}, \boldsymbol{p}$.
3: **Initialize:** Set $\mathcal{U}_c, \mathcal{U}_d, \mathcal{N}$, $\lambda = 300$, $\epsilon = 0.01$, $l_a = 100$ and $l_b = 1$.
4: **Step 1:** Determine user association $\boldsymbol{x}$.
5: **for** $m = 1$ to M **do**
6:   **if** m is inside range λ **then**
7:     Set $x_{m0} = 1$ and update $\mathcal{U}_c = m$.
8:   **else**
9:     For TD m, find D2D RD k within a distance $d^{\max}$.
10:     Set $x_{mk} = 1$ and update $\mathcal{U}_d = m$.
11:   **end if**
12: **end for**
13: **Step 2:** Based on x decide computation resource allocation.
14: $\forall m \in \mathcal{U}_c$, allocate computation resource as $f_{m0} = F_0/|\mathcal{U}_c|$.
15: $\forall m \in \mathcal{U}_d$, allocate computation resource as $f_{mk} = F_k$.
16: **Step 3:** Initialize $\alpha_m = \alpha_m^{\min}$, $\forall m \in \mathcal{M}$ using (13).
17: **Step 4:** Iteratively adjust data offloading ratio to reduce the parallel processing lag
18: **while** $l_a - l_b > \epsilon$ **do**
19:   **for** $m = 1$ to M **do**
20:     **if** $\alpha_m < \alpha_m^{\max}$ **then**
21:       Update $\alpha_m = \alpha_m + 1$.
22:     **end if**
23:     Determine $R_m^{\min}$ using (16).
24:   **end for**
25:   Allocate communication resources to all cellular TDs using Algorithm 1.
26:   Allocate communication resources to all D2D TDs using Algorithm 2.
27:   Set $l_a = \bar{L}_m^{\text{loc}}$.
28:   Set $l_b = \bar{L}_m^{\text{off}} + \bar{L}_m^{\text{rem}}$.
29: **end while**

B. JPORA

Here, we discuss the overall scheme which involves user association, data segmentation, and resource allocation to minimize the total latency of D2D enabled computation offloading network. The global algorithm for our scheme is presented in Algorithm 3. In general, the global algorithm consists of four main steps: user association, computation resource allocation for remote execution, initial data segmentation decision, and iterative reduction of the parallel processing delay.

- *Step 1:* The first step is to decide the user association to determine the cellular and D2D links and subsequently make the computation and communication resource allocation decisions. The user association problem is solved based on the location of a TD as described in Section V. In Line 7, if a TD is within the range λ, it is associated to the SBS via cellular link. Hence, the cellular links set is updated. Otherwise, D2D offloading is decided for a TD. In Line 10, one-to-one D2D association is determined between the TD m and D2D RD k such that the distance between them is not greater than $d^{\max}$, and the D2D links set is updated accordingly.
- *Step 2:* In the second step, the computation resource allocation for remote execution is decided. As the user association is known, the cellular TDs are allocated computation resources at the SBS such that the finite computation capacity constraint of the SBS in (11l) is satisfied. In Line 14, the computation resources of the SBS are uniformly distributed among all the associated TDs. On the other hand, the computation capacity of a D2D RD denoted as F_k is allocated to the associated TD.
- *Step 3:* In the third step, the offloading fraction is initialized to minimum value of α_m derived in Section V-A by taking into account the constraint on local computation latency in (11b) which bounds the local execution time by the maximum delay threshold.
- *Step 4:* The fourth step is the main step of our algorithm, which aims to reduce the lag between the two parallel processes of local computation and computation offloading by adjusting the offloading fraction at each TD in an iterative manner with accuracy ϵ. Initially the value of offloading fraction is set to $\alpha_m^{\min}$, which implies that the local execution time is always bounded within the delay threshold. However, the delay constraint may not be satisfied for computation offloading process initially. Therefore, the while loop in Line 18 adjusts the data segmentation in a way that the two parallel processes take the same time, which in turn ensures the latency constraint for computation offloading as well.

In each iteration, the data segmentation for each TD is adjusted in Line 21 within the upper bound $\alpha_m^{\max}$, which ensures that the local energy consumption constraint in (11c) is satisfied. In Line 23, the minimum data rate corresponding to the offlaoding fraction value is estimated. Then QoS-aware communication resource allocation for cellular TDs is performed using Algorithm 1, followed by the interference-aware subcarrier assignment and power allocation for D2D TDs using Algorithm 2. Based on the obtained offloading fraction, communication and computation resource allocation, the corresponding local computing, offloading and remote computing latencies can be computed. The iteration continues until the difference between local computing and computation offloading becomes close to the value of ϵ. Instead of reducing the latency for each TD sequentially, the condition for while loop is updated based on average local computing latency and average computation offloading latency for all the TDs in Lines 27 and 28.

The computation complexity of proposed JPORA scheme mainly lies in the iterative step, which uses Algorithm 1 and Algorithm 2. The number of iterations of while loop is determined by the accuracy ϵ. Given $\epsilon > 0$, the complexity of one dimensional search on partial offloading fraction is $O(log(1/\epsilon))$. Assuming $N >> M$, the complexity of Algorithm 1 dominates in each iteration, and thus the worst case computation complexity of JPORA scheme is given as $O(N^2 M log(1/\epsilon))$.

VII. Performance Evaluation

In this section, we evaluate the performance of our proposed scheme by analyzing the numerical results and comparing with the following four benchmark schemes:

1) *Local Computation:* Each TD computes its task locally and there is no offloading involved.
2) *Random Offloading:* The fraction of task data to be offloaded at each TD is decided randomly, while user association is same as our scheme. The communication and computation resources are distributed uniformly, whereas the reuse pairs for spectrum sharing are decided at random.
3) *Complete Offloading:* Each TD offloads its task completely for remote execution, while user association is same as our scheme. The communication and computation resource allocation is performed similar to random offloading scheme.
4) *LPORA:* Similar to the existing work in [21], each TD offloads part of its task for remote execution only if it cannot be completed on time by local computing. Hence, the size of computation executed locally at TD m is $\tau_m f_m / C_m$, while remaining is offloaded to the associated RD. As the authors in [21] overlooked interference by considering dedicated spectrum, the communication resource allocation in LPORA is adopted from our proposed scheme for fair comparison.

We consider a BS with radius 500 m and randomly located TDs. The total bandwidth is divided into $N = 64$ orthogonal subcarriers. The channel gain h_m^n is modeled as independent Rayleigh fading channel which incorporates the path loss and shadowing effects [42]. For user association, the radius λ is set to 300 m, while the maximum distance for D2D association is set as $d^{\max} = 50$ m. The expected energy consumption for each TD is randomly chosen from $\{1, 2\}$ J, while the local computing energy per cycle $\mathcal{W}$ follows a uniform distribution in the range $(0, 20 \times 10^{-11})$ J/cycle similar to [13], [12]. The local computation capacity f_m of TD follows uniform distribution between [0.5, 1] GHz. The computation capacities of D2D RDs and the SBS are set as $F_k = 2.5$ GHz and $F_0 = 25$ GHz, respectively. The accuracy is set as $\epsilon = 0.01$. The other key parameters specific to network and computation task are listed in Table II unless stated otherwise.

TABLE II
Simulation Parameters

Parameter	Value
Bandwidth B	20 MHz
Noise power N_0	-173 dBm
Maximum transmit power of TD $P_m^{\max}$	200 mW
Task size D_m	[100,500] KB
Processing density C_m	[1000,1500] cycles/bit
Delay threshold τ_m	500 msec
Expected energy consumption E_m	{1,2} J

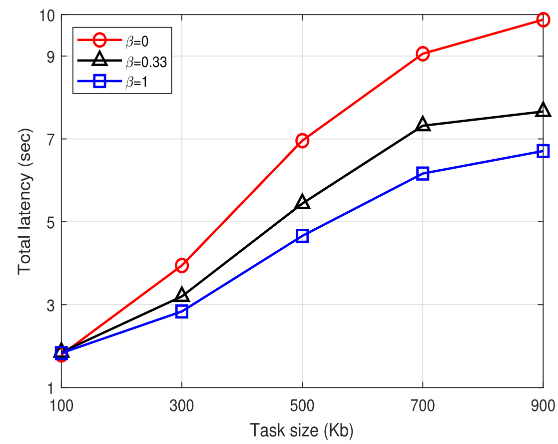


Fig. 2. Effect of D2D to cellular TDs ratio on total latency.

We proposed D2D computation offloading to reduce the load on SBS, and enhance the latency performance by enabling proximal task offloading. Therefore, it is important to verify the performance gain contributed by D2D task offloading. In this context, we define the ratio of D2D to cellular TDs in the network as β. In Fig. 2, we plot total latency against the task size for three different cases of β: 1) $\beta = 0$ (no D2D offloading) 2) $\beta = 0.33$ (one fourth of the TDs perform D2D offloading) 3) $\beta = 1$ (equal number of cellular and D2D offloading TDs). It can be observed from the figure that, the total latency increases with the increase in the data size. It is mainly due to limited bandwidth and finite edge computation resources, which impact the transmission and remote execution delays, respectively. Comparison among the curves shows that, the minimum latency is achieved when there are equal number of D2D and cellular TDs corresponding to $\beta = 1$, while the latency performance degrades as the cellular

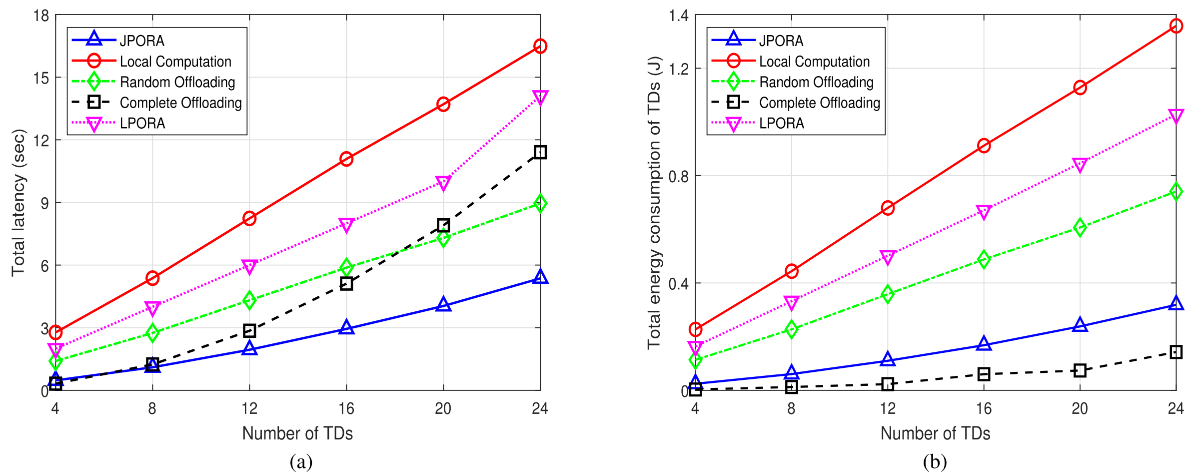


Fig. 3. Effect of increasing number of TDs on total latency and total local energy consumption for all the TDs, where the number of D2D and cellular TDs is equal ($\beta = 1$). (a) Total latency vs number of TDs. (b) Total local energy consumption vs number of TDs.

TDs outnumber the D2D TDs. For $\beta = 0$, the latency performance is the worst as all the TDs offload computation to the SBS irrespective of their channel conditions and the finite computing capacity of the SBS.

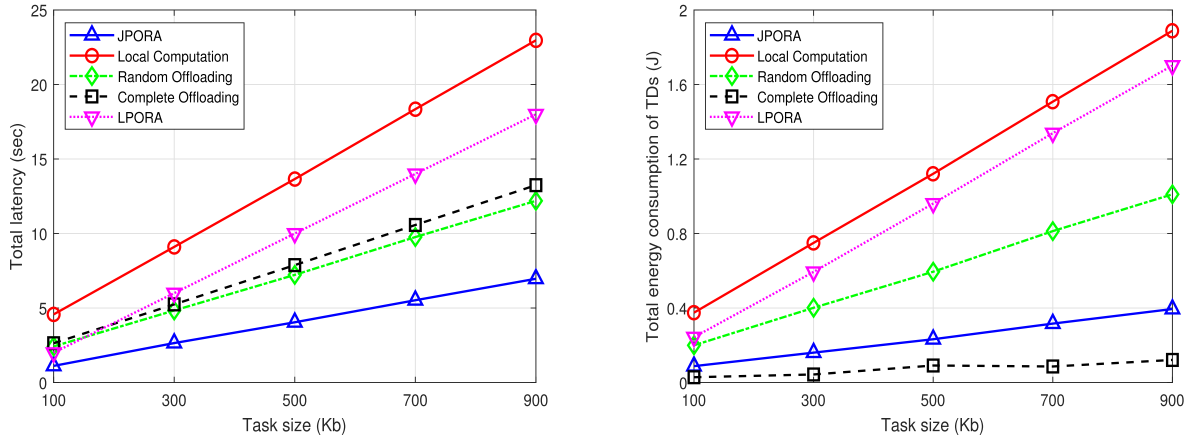
Practically, the computation and communication load (in terms of computation offloading requests) can vary at large with the increase in user density. In this context, we need to verify the scalability and robustness of the proposed scheme. Therefore, in Fig. 3, we analyse the performance of our scheme by varying the number of TDs or alternatively the offloading requests, while the number of cellular and D2D TDs is fixed as ($\beta = 1$). Specifically, Fig. 3(a) plots total latency and shows an increasing trend with increase in number of TDs for all the schemes. It is obvious, as with increase in cellular and D2D offloading users the bandwidth and computing resources allocated to each user decrease. It is interesting to note that, complete offloading and JPORA achieve almost same latency for less number of TDs, which is due to sufficient resources at the SBS. However, increase in TDs leads to resource contention at the SBS due to which partial offloading schemes achieve better performance. Comparison among all the schemes shows that JPORA has much smaller increasing rate. In particularly, JPORA achieves approximately 60% lower latency than LPORA, which alone substantiates the effectiveness of our partial offloading strategy that efficiently utilizes the local and remote resources to reduce the execution delay. On the other hand, JPORA performs approximately 40% better than random offloading due to both effective data segmentation and interference-aware radio resource allocation. As compared to local computation and complete offloading schemes, JPORA decreases the latency approximately by 67% and 50%, respectively.

Fig. 3(b) plots the total energy consumption of the TDs, which is sum of the energy consumed by each TD in local execution and data transmission. As the number of TDs increase, the energy consumed in local execution and communication by each TD adds to the total mobile energy consumption. JPORA has lowest increasing rate, and decreases the energy consumption by 70% and 53% as compared to random offloading and LPORA, respectively. This is mainly due to the fact that, the data segmentation strategy in JPORA is derived under the desired local energy consumption of TD. Moreover, proper interference management while offloading the task data results in lower power consumption. However, complete offloading has lowest energy consumption as it avoids the local execution cost.

From Fig. 3, it is evident that, the proposed scheme can significantly reduce the latency within the delay threshold while ensuring the desired local energy consumption of TDs as the network becomes dense. For instance, when number of TDs is 24 the average latency is approximately 270 msec, while the delay threshold is 500 msec, which means a decrease of 46%.

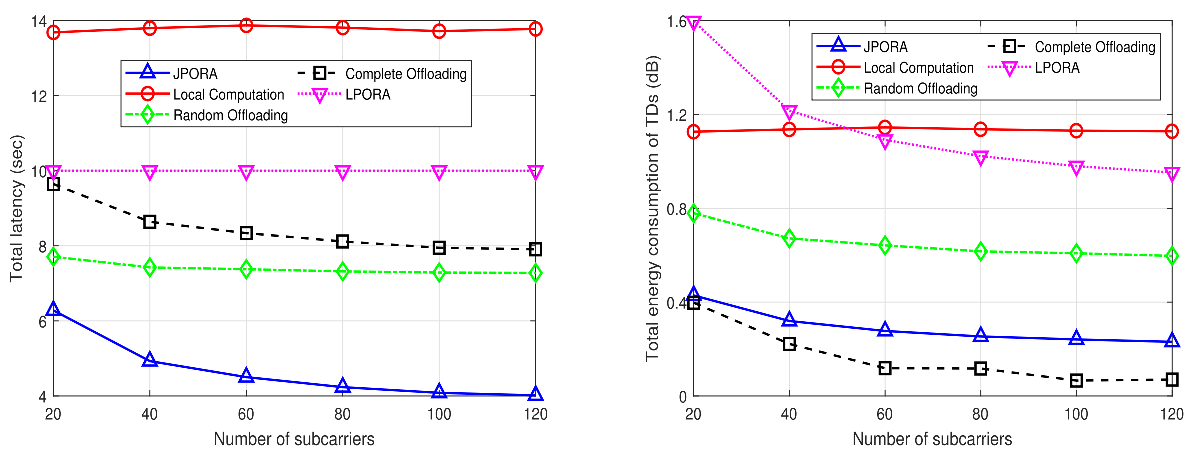
Keeping in view the variation in the computation task size due to diversity in user applications, we verify the performance of JPORA for increasing task size in Fig. 4. For each data point the task size is uniformly distributed between 100 KB and the task size value on that data point. Fig. 4(a) plots the total latency, and shows that it gradually increases with increase in data size for all the schemes. This can be explained as, with the increase in task size higher fraction of data is offloaded for remote execution as the computation capacity and desired local energy consumption for a TD is limited, which in turn leads to higher transmission delay in a bandwidth limited scenario. Also, the remote execution time increases when the offloaded task size is increased. Comparison shows that, JPORA has lowest increasing rate. Moreover, it reduces the total latency by approximately 70%, 42%, 46%, and 63% than local computation, random offloading, complete offloading, and LPORA schemes, respectively. This performance gain is mainly attributed to the integrated partial offloading and interference management of our proposed scheme.

Fig. 4(b) depicts the increasing trend of energy consumption, as increase in input task size not only overburdens the TD with extra processing but also exhausts the communication resources. As compared to local computation, random offloading, and LPORA, our proposed scheme consumes approximately 79%, 60%, and 76% less energy and has lowest increasing rate. On



(a) (b)

Fig. 4. Effect of increase in task size on total latency and total local energy consumption for all the TDs. (a) Total latency vs task size. (b) Total local energy consumption vs task size.



(a) (b)

Fig. 5. Effect of increase in bandwidth on total latency and total local energy consumption for all the TDs. (a) Total latency vs number of subcarriers. (b) Total local energy consumption vs number of subcarriers.

the other hand, complete offloading achieves minimum energy consumption as local execution is not involved.

Bandwidth resources in networks are generally elastic and their variation can impact the offloading performance in a shared spectrum. As the communication resource allocation is a challenging part of our problem, we need to determine the operating bandwidth to verify the application of our scheme in practical scenarios. Therefore, we analyse the effect of increase in available bandwidth in terms of number of subcarriers on the total latency and local energy consumption in Fig. 5, and provide insight into the optimal communication resource allocation. Here, we assume that the total bandwidth is variable and fix the bandwidth per subcarrier as $\bar{B} = 180$ KHz. Fig. 5(a) plots the total latency against the increasing number of subcarriers and shows a slight decrease in latency for our scheme. The reduction in delay is contributed by the increased bandwidth since more bandwidth resources enable higher data rate, which reduces the transmission delay. However, increasing the bandwidth beyond a certain range does not reduce the latency any further, as it can be observed from almost flat curves of random offloading and complete offloading. Although, LPORA employs same resource allocation as our scheme, its performance does not improve due to its data segmentation strategy which always leads to higher local execution delay. For maximum number of subcarriers, JPORA can achieve 71%, 49%, 50%, and 60% less latency than other four baseline schemes, respectively.

Fig. 5(b) plots the total energy consumption of TDs vs the subcarriers. For the proposed scheme, the total energy consumption decreases slowly with the increase in radio resources due to the fact that more subcarriers are available per TD, resulting in lower transmission power and lower respective energy consumption. Although transmission energy consumption is same for JPORA and LPORA, the local energy consumption is higher in case of LPORA due to inefficient partial offloading strategy. Moreover, the increase in subcarriers has relatively much less impact on the performance of complete and random offloading schemes as these schemes fail to manage cross-tier interference.

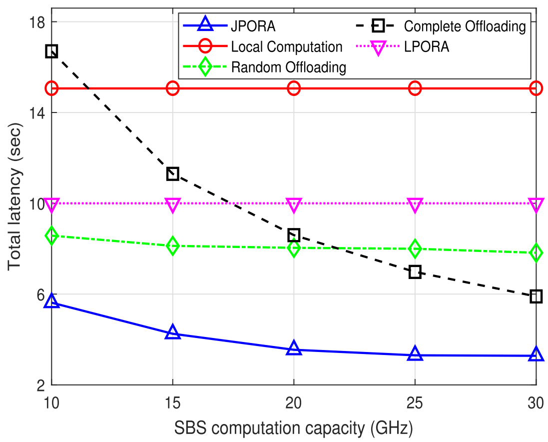


Fig. 6. Effect of SBS computation capacity on total latency.

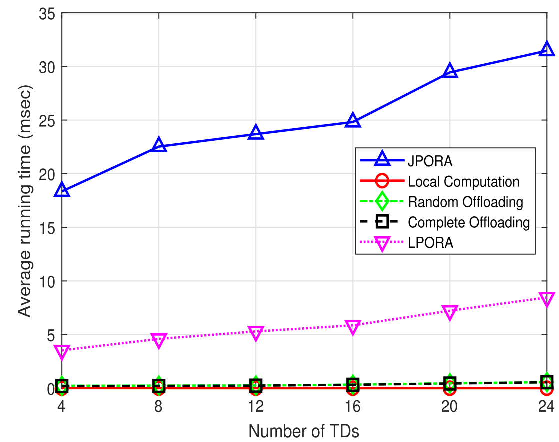


Fig. 7. Comparison of average running time against number of TDs for different baseline schemes.

It can be deduced from the results in Fig. 5 that for a network setting with 20 TDs, increasing the bandwidth beyond a certain value does not improve the total latency and energy performance any further. The threshold value for JPORA is around 18 MHz corresponding to $N = 100$ subcarriers of 180 KHz each, while maximum uplink bandwidth realised by LTE system is 20 MHz.

In order to analyse the impact of edge resources on the performance of JPORA, we plot the total latency against computation capacity of the SBS in Fig. 6. The figure shows different trends for the four baslines. The latency in case of local computing is disregarded for comparison as it is independent of SBS computation capacity. The invariant performance for random offloading and LPORA schemes can be explained as even the remote execution latency improves, the local execution delay is higher and the latency in our objective is the maximum of these two delays. In case of complete offloading the latency reduces sharply as the resource allocated per user increases with increase in edge capacity. For JPORA, the total latency decreases initially, however, the performance does not improve any further beyond a certain threshold as the SBS computation capacity does not remain the dominant constraint. It is important to note that, the edge capacity beyond 25 GHz is hard to realise in practice. For edge capacity of 25 GHz, JPORA provides 79%, 61%, 55% and 71% lower latency than the other four baseline schemes, respectively.

Finally, we provide a comparison of the computational complexity of each scheme running on a computer with Intel Core i7-6700 K 4 GHz CPU and 32 GB RAM. Fig. 7 plots average running time per iteration by increasing the number of TDs. With the increase in number of TDs, the running time increases gradually for JPORA. For maximum number of TDs, JPORA can find the solution in less than 35 msec. As compared to other baselines, the higher computational complexity of JPORA is obvious due to the iterative approach for balancing the local and remote execution processes. Although the run time complexity of JPORA is highest among all the schemes, it is considerably less than the required delay threshold.

VIII. Discussion

The above performance evaluation proofs that JPORA can effectively reduce the latency meanwhile preserving local energy consumption for joint MEC and D2D computation offloading in a shared spectrum. The performance comparison under various key parameters demonstrates that our proposed scheme outperforms other baseline schemes, and can scale well to increased offloading demand and varying application sizes. Moreover, our analysis also provides the operating thresholds for JPORA, for instance maximum load in terms of users, effective bandwidth, and effective edge capacity.

Nevertheless, there are some limitations from practical implementation aspect. First, there is a cost of partially processing the task, which needs to be considered while formulating the partial offloading based latency minimization problem. To address this limitation in our future work, we intend to formulate cost as our objective function by jointly considering the cost of execution time and task partitioning.

Second, the D2D-enabled MEC framework substantially relies on devices' collaboration. Therefore, effective incentive mechanism to prevent overutilization and free-riding behaviors is highly desirable. Keeping that in view, we aim to integrate social-tie based incentive mechanism for D2D collaboration similar to the existing work in [43]. The key idea is to leverage social trust and social reciprocity for trust worthy communication and computation resource sharing among proximal devices. As an initial attempt, we briefly discuss the idea of integrating this incentive mechanism in our framework. A multilayered graph can be constructed with D2D connections and social links. Then initial offloading decision or user association can be determined by taking into account the device social graph and D2D connectivity graph.

IX. Conclusion

In this paper, we investigated the total latency minimization problem for a D2D-enabled partial computation offloading scenario in an OFDMA system, meanwhile taking into account the spectral efficiency. We jointly considered user association, partial offloading, and resource allocation to minimize the total latency. We proposed JPORA scheme which iteratively reduced the parallel processing delay by adjusting the data segmentation and solving the underlying key challenge of interference in a shared spectrum. The simulation results demonstrated that our proposed scheme achieved better latency and energy performance than other baseline schemes.

As a future work, we will consider the task partitioning cost as a part of our objective function. Also, we intend to design an incentive mechanism to motivate the D2D cooperation among devices for computation offloading and communication resource sharing. Specifically, we will investigate latency minimization problem in D2D integrated MEC scenario by considering the social trust and social reciprocity based cooperative D2D communications.

REFERENCES

[1] M. Weldon, “Nokia Bell Labs: The rational exuberance of 5G,” 2016. [Online]. Available: https://www.bell-labs.com/var/articles/rational-exuberance-5G

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] R. He *et al.*, “Propagation channels of 5G millimeter wave vehicle-to-vehicle communications: Recent advances and future challenges,” *IEEE Veh. Technol. Mag.*, vol. 15, no. 1, pp. 16–26, Mar. 2019.

[4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing a key technology towards 5G,” *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[5] L. Zhao, J. Wang, J. Liu, and N. Kato, “Optimal edge resource allocation in IoT-based smart cities,” *IEEE Netw.*, vol. 33, no. 2, pp. 30–35, 2019.

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

[7] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, “Delay-optimal computation task scheduling for mobile-edge computing systems,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2016, pp. 1451–1455.

[8] 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.

[9] 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.

[10] H. Q. Le, H. Al-Shatri, and A. Klein, “Efficient resource allocation in mobile-edge computation offloading: Completion time minimization,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 2513–2517.

[11] J. Ren, G. Yu, Y. Cai, Y. He, and F. Qu, “Partial offloading for latency minimization in mobile-edge computing,” in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–6.

[12] U. Saleem, Y. Liu, S. Jangsher, and Y. Li, “Performance guaranteed partial offloading for mobile edge computing,” in *Proc. IEEE Global Commun. Conf.*, 2018, pp. 1–6.

[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. 2016.

[14] D. Chatzopoulos, C. Bermejo, E. Ul Haq, Y. Li, and P. Hui, “D2D task offloading: A dataset-based Q&A,” *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 102–107, Feb. 2019.

[15] J. Liu, N. Kato, J. Ma, and N. Kadowaki, “Device-to-device communication in LTE-advanced networks: A survey,” *IEEE Commun. Surveys Tut.*, vol. 17, no. 4, pp. 1923–1940, Oct.–Dec. 2014.

[16] J. Liu, N. Kato, H. Ujikawa, and K. Suzuki, “Device-to-device communication for mobile multimedia in emerging 5G networks,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 5s, 2016.

[17] J. Liu, S. Zhang, N. Kato, H. Ujikawa, and K. Suzuki, “Device-to-device communications for enhancing quality of experience in software defined multi-tier LTE-A networks,” *IEEE Netw.*, vol. 29, no. 4, pp. 46–52, Jul.–Aug. 2015.

[18] J. Liu, H. Nishiyama, N. Kato, and J. Guo, “On the outage probability of device-to-device-communication-enabled multichannel cellular networks: An RSS-threshold-based perspective,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 163–175, Jan. 2016.

[19] J. Liu, Y. Kawamoto, H. Nishiyama, N. Kato, and N. Kadowaki, “Device-to-device communications achieve efficient load balancing in LTE-advanced networks,” *IEEE Wireless Commun.*, vol. 21, no. 2, pp. 57–65, Apr. 2014.

[20] X. Chen and J. Zhang, “When D2D meets cloud: Hybrid mobile task offloadings in fog computing,” in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[21] Y. He, J. Ren, G. Yu, and Y. Cai, “D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1750–1763, Mar. 2019.

[22] N. T. Ti and L. B. Le, “Computation offloading leveraging computing resources from edge cloud and mobile peers,” in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[23] R. Chai, J. Lin, M. Chen, and Q. Chen, “Task execution cost minimization-based joint computation offloading and resource allocation for cellular D2D MEC systems,” *IEEE Syst. J.*, vol. 13, no. 4, pp. 4110–4121, Dec. 2019.

[24] G. Hu, Y. Jia, and Z. Chen, “Multi-user computation offloading with D2D for mobile edge computing,” in *Proc. IEEE Global Commun. Conf.*, 2018, pp. 1–6.

[25] C. Wang, J. Qin, X. Yang, and W. Wen, “Energy-efficient offloading policy in D2D underlay communication integrated with MEC service,” in *Proc. 3rd Int. Conf. High Performance Compilation, Comput. Commun.*, 2019, pp. 159–164.

[26] J. Dai, J. Liu, Y. Shi, S. Zhang, and J. Ma, “Analytical modeling of resource allocation in D2D overlaying multihop multichannel uplink cellular networks,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 6633–6644, Aug. 2017.

[27] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, “AC-POCA: Anticoordination game based partially overlapping channels assignment in combined uav and D2D-based networks,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1672–1683, Feb. 2018.

[28] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Commun. Surveys Tut.*, vol. 19, no. 4, pp. 2322–2358, Oct.–Dec. 2017.

[29] P. Mach and Z. Becvar, “Mobile edge computing: A survey on architecture and computation offloading,” 2017, *arXiv:1702.05309*.

[30] J. Wang, L. Zhao, J. Liu, and N. Kato, “Smart resource allocation for mobile edge computing: A deep reinforcement learning approach,” *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: 10.1109/TETC.2019.2902661.

[31] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, “Mobile-edge computation offloading for ultradense IoT networks,” *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4977–4988, Dec. 2018.

[32] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, “Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration,” *IEEE Trans. Comput.*, vol. 67, no. 9, pp. 1287–1300, Sep. 2018.

[33] J. Liu, Y. Shi, L. Zhao, Y. Cao, W. Sun, and N. Kato, “Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 221–232, Feb. 2018.

[34] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, “Hybrid method for minimizing service delay in edge cloud computing through vm migration and transmission power control,” *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2016.

[35] Q. -V. Pham, T. LeAnh, N. H. Tran, and C. S. Hong, “Decentralized computation offloading and resource allocation in heterogeneous networks with mobile edge computing,” 2018, *arXiv:1803.00683*.

[36] M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma, and J. Ott, “Security and privacy in device-to-device (D2D) communication: A review,” *IEEE Commun. Surveys Tut.*, vol. 19, no. 2, pp. 1054–1079, Apr.–Jun. 2017.

[37] O. Munoz, A. P.-Iserte, and J. Vidal, “Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2014.

[38] D. Pisinger, “Where are the hard knapsack problems?” *Comput. Oper. Res.*, vol. 32, no. 9, pp. 2271–2284, 2005.

[39] C. Chekuri and S. Khanna, “A PTAS for the multiple knapsack problem,” *Departmental Papers (CIS)*, p. 146, 2005.

[40] M. Li, S. Yang, Z. Zhang, J. Ren, and G. Yu, “Joint subcarrier and power allocation for OFDMA based mobile edge computing system,” in *Proc. IEEE 28th Annu. Int. Symp. Personal, Indoor, Mobile Radio Commun.*, 2017, pp. 1–6.

[41] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 2714–2741, Oct.–Dec. 2018.

[42] K. Kim, Y. Han, and S.-L. Kim, "Joint subcarrier and power allocation in uplink OFDMA systems," *IEEE Commun. Lett.*, vol. 9, no. 6, pp. 526–528, Jun. 2005.

[43] X. Chen, B. Proulx, X. Gong, and J. Zhang, "Exploiting social ties for cooperative D2D communications: A mobile social networking case," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1471–1484, Oct. 2015.

Umber Saleem received the B.E. degree in information and communication systems engineering in 2013 and the M.S. degree in electrical engineering (telecommunications) in 2016 from the National University of Sciences and Technology, Islamabad, Pakistan. She is currently working toward the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China. Her research focus is resource allocation for mobile edge mobile edge computing.

Yu Liu received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 2018. He is currently working toward the Ph.D. degree in electronic engineering with Tsinghua University, Beijing, China. His research interests include wireless networks, edge computing, and optimization.

Sobia Jangsher (Member, IEEE) received the B.E. degree in electronics engineering and the M.S. degree in communication system engineering from the National University of Science and Technology, Islamabad, Pakistan and the Ph.D. degree in wireless communication from The University of Hong Kong, Hong Kong. She did her M.S. thesis on "Adaptive transmission of video over MIMO channels" under the supervision of Dr. Syed Ali Khayam and the Ph.D. thesis on "Resource Allocation in Moving Small Cell Network" under the supervision of Prof. Victor O.K Li. She is currently working as an Assistant Professor with the Institute of Space Technology, Islamabad, Pakistan. Her research mainly focuses on resource allocation in future wireless communication systems.

Xiaoming Tao, Photograph and biography not available at the time of publication.

Yong Li (Senior Member, IEEE) received the B.S. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007 and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012. He is currently a Faculty Member with the Department of Electronic Engineering, Tsinghua University.

Dr. Li has served as General Chair, TPC Chair, SPC/TPC Member for several international workshops and conferences, and he is on the editorial board of two IEEE journals. His papers have total citations more than 6900. Among them, ten are ESI Highly Cited Papers in Computer Science, and received four Conference Best Paper (runer-up) Awards. He received IEEE 2016 ComSoc Asia-Pacific Outstanding Young Researchers, Young Talent Program of China Association for Science and Technology, and the National Youth Talent Support Program.