NOMA-Enabled Computation and Communication Resource Trading for a Multi-User MEC System

Sabyasachi Gupta, Dinesh Rajan, *Senior Member, IEEE*, and Joseph Camp, *Member, IEEE*

***Abstract*—In this article, we establish a novel framework for a multi-user mobile edge computing (MEC) network in which a set of users with high downlink rate demands and a set of users with intensive computation tasks can collaborate to achieve a mutually-beneficial scenario such that completion time of the tasks is reduced and the base station (BS) can send more information at a higher rate to the downlink users. Specifically, by leveraging non-orthogonal multiple access (NOMA) for uplink and downlink traffic, the user with the computation task can offload shares of the computation task to the edge cloud and the downlink user. At the same time, this user forwards the information it receives from the BS to the downlink user. In this set up, we jointly optimize the communication resources, computational resources at the edge cloud and user devices, pairings among the two sets of users, the shares of computation tasks, and relay bits to minimize the total task completion time while satisfying downlink users' incentive requirements. For a network with a single computation demanding user and a single downlink user, the optimal solution to the problem is provided. For a network with multiple users, the problem is non-convex and computationally challenging. Hence, we propose an efficient, low complexity algorithm that utilizes the bottleneck matching algorithm, convex optimization, and the block coordinate descent scheme to obtain a locally-optimal solution. Simulation results demonstrate that, as compared with the state-of-the-art, the total task completion time is greatly reduced (32%–51%), and a large computational energy savings at the edge cloud (38%–55%) is achieved. Simultaneously, the downlink users' rates improve compared to the orthogonal transmission.**

***Index Terms*—Incentive design, mobile-edge computing, resource allocation, task offloading.**

I. Introduction

With the ever-increasing utilization of mobile devices, advanced applications with high computational load (*e.g.*, fingerprinting or face recognition, natural language processing, virtual reality, and interactive gaming) are becoming popular. Mobile-edge computing (MEC) has emerged as a promising solution to address the computation demands of these applications. MEC technology provides computing facilities and storage resources at the network edge, *e.g.* base stations (BSs), with aim of minimizing the service latency of these applications [2].

As the number of computing user equipments (CUEs) that require computing latency-sensitive tasks is growing, offloading all the tasks to the edge cloud with finite computational resources may not always be advantageous. Over the last decade, mobile devices' computational power has increased steadily to where the computational performance of a mid-range mobile processor is 10% that of a mobile edge server processor (*e.g.*, Intel Xeon D-Series) [3]. One or more mobile devices with an idle processor may be present on a wireless network. Therefore, offloading CUE's computation-intensive tasks to these peers is an appealing choice. The challenge with this approach is that encouraging other users to help in computing the task of a CUE may prove to be complicated since users in the network may be selfish. Therefore, appropriate incentives are required for these helping mobile peers.

We consider a network with two different sets of users that have varied communication and computation requirements: (*i*.) A set of downlink users (DUEs) having high data rate demands and underutilized computation resources. For example, the DUEs may be interested in experiencing high-quality multimedia content, such as with virtual reality (VR). (*ii*.) A set of CUEs with computationally-intensive tasks with stringent completion times, having high throughput (*i.e.*, low task offloading delay) and underutilized channels with the BS. In such cases, the task computation time is much higher than the task offloading delay. For such a network, we design a framework in which a CUE can trade its communication resources (*e.g.*, offloading delay and communication energy) for computation resources from a DUE, which leads to a mutually-beneficial scenario, where the DUE can receive information from the BS at higher rates,[1] while the CUE's task completion time can be greatly reduced. In this case, the BS sends downlink information intended for the DUE via non-orthogonal multiple access (NOMA) to both the CUE and DUE. The CUE offloads a share of its computation bits to the edge cloud and another share of its computation bits along with the information bits that it receives from the BS to the DUE using NOMA. The excess information bits (compared to the orthogonal transmission through the BS to DUE link) that a DUE receives is the main incentive for the DUE to compute the CUE's

Manuscript received May 20, 2021; revised November 27, 2021 and March 3, 2022; accepted April 12, 2022. Date of publication April 19, 2022; date of current version July 18, 2022. This work was supported in part by NSF under Grants CNS-1823304 and CNS-1909381, and in part by the Air Force Office of Scientific Research under Grant FA9550-19-1-0375. This work was presented in part at IEEE WCNC, Nanjing, China, March 2021 [1]. The review of this article was coordinated by Prof. Hung-Yun Hsieh. *(Corresponding author: Sabyasachi Gupta.)*

The authors are with the Department Electrical and Computer Engineering, Southern Methodist University, Dallas, TX 75275 USA (e-mail: sabyasachig@smu.edu; rajand@smu.edu; camp@smu.edu).

Digital Object Identifier 10.1109/TVT.2022.3168503

[1]A high quality VR experience could be achieved, even with a low-to-moderate increase in transmission rate, by utilizing a scalable video coding technique and predicting the relevant portion of the 360° video that the user may need in the next time slot [4], [5].

task. For a network with multiple CUEs and DUEs, we aim to minimize the overall task completion time of all the CUEs while maintaining each participating DUE's incentive requirement, energy constraint at each CUE, and finite computation resource at the edge cloud. The major contributions of this paper are:

- We propose a novel framework that allows for a mutually-beneficial trade-off between the computation and communication resources between CUEs and DUEs. By enabling NOMA in the uplink and downlink directions, the completion time for a CUE's task can be reduced, and the DUE can receive information from the BS at a higher transmission rate. Furthermore, by efficiently designing the transmission protocol, we show that interference-free transmission rates in the CUE-DUE and CUE-BS links can be achieved simultaneously under a certain channel condition.
- We design each participating DUE's utility based on two factors: the additional bits received compared to the orthogonal transmission and the extra energy consumption to compute the task for a CUE. Each DUE can specify a parameter, the trading factor, that quantifies the reward (in terms of increased data rate) that it expects for each bit of computation that is provided to the CUE. We include a constraint that the DUE is willing to cooperate with the CUE if it can achieve a non-negative utility.
- For a network with a single CUE and a single DUE, we optimally solve the CUE's task completion time minimization problem. For a multi-user network, we consider the problem of minimizing the maximum task completion time among all CUEs. The design parameters are: the selection of the beneficial DUE for each CUE, the edge cloud's computational resource allocation for each CUE, computation resource allocation (*i.e.*, computation time) at each user device, communication resource allocation, the share of computation, and relay bits. The optimization problem is non-convex. Therefore, an iterative algorithm is proposed that can solve the optimization problem with significantly lower computational complexity. It is shown that the algorithm converges, at least to a local optimal solution.
- We compare our proposed method with two state-of-the-art methods: (*i.*) computational offloading to the edge cloud using orthogonal transmissions, and (*ii.*) computational offloading using uplink NOMA. We show that using the proposed technique, CUEs' task completion times are greatly reduced (32%–51%) compared to the state-of-the-art methods. Furthermore, DUEs can achieve large bit gains (compared to the orthogonal transmission). We have also shown that a large computation energy savings (38%–55%) at the edge cloud is achieved using the proposed technique.

The rest of this paper is organized as follows. In Section II, we describe related work. In Section III, we describe the proposed system model. In Section IV, we formulate the optimization problem. In Section V, we introduce the proposed solution method. In Section VI, we evaluate our results. Finally, we conclude this article in Section VII.

II. Related Work

Designing the communication and computation resource allocation and computation share among different devices in an MEC network has attracted significant attention over the last few years. Energy consumption and latency are commonly considered when evaluating the performance of an MEC system. In [6], joint optimization of computation and communication resources was investigated for an MEC network with a single edge cloud. The objective was to minimize weighted sum energy consumption. For a multi-user network, the joint optimization problem of computation task sharing and offloading time allocation with the aim of minimizing the completion time of the computing users was investigated in [7]. In [8], joint optimization of computation resource, communication resource, and task offloading decisions were considered for an MEC network with multiple edge clouds. The objective function was defined as a weighted sum of the task completion time and energy consumption.

In spite of recent progress on resource optimization for MEC networks, scaling to a large number of users is challenging since the computational capability of the edge cloud is finite. If a large number of users offload their tasks to the edge cloud, a significant computational delay may be observed at the edge cloud since per-user computational resource availability at the edge cloud is greatly reduced. A computing node may be surrounded by mobile peers with idle processors. Therefore, in [9]–[12], task offloading to mobile peers was considered, and joint mobile peer selection and computational resource allocation problems were studied. For a network in which each user has a computing task, He *et al.* [9] investigated the optimization problem of maximizing the number of users that successfully complete their tasks within a specified deadline by offloading to a peer user and the edge cloud. In [10], joint optimization of computation and communication resources was investigated for an MEC system in which the task was computed with the help of a peer device and an edge cloud. The objective was to minimize energy consumption while satisfying the user's task completion time constraint. In [11], a multi-helper MEC system was investigated in which a local user can offload its computation task to multiple nearby devices using time division multiple access (TDMA). The overall delay of the MEC network was minimized by optimizing computation and communication delays as well as the task offloading decision. It has been assumed in [9]–[12] that helping mobile devices are willing to compute the tasks for the CUEs without receiving any incentive, which may not be the case if these devices have limited battery energy. In [13], an MEC system was investigated in which computation shares are offloaded to neighboring mobile devices. Each helping mobile device received a bandwidth incentive to compute the task. For vehicular networks, computation offloading to the neighboring vehicles and monetary incentives for the helping vehicles was considered in [14], [15].

Recently, NOMA has been recognized as one of the key approaches in fifth-generation (5G) networks. NOMA allows multiple users to share the same resource (*e.g.*, frequency, or time) unit channel for simultaneous transmissions using

successive interference cancellation (SIC), and therefore, it can achieve higher spectral efficiency compared to orthogonal multiple access (OMA). Motivated by the advantages of NOMA over OMA, a large body of works investigated NOMA for MEC networks [16]–[25]. Ding *et al.* [17] investigated power allocation and selection of the best mode among OMA and NOMA schemes with the aim of minimizing offloading delay for two-user MEC network. In [18], to minimize computation offloading energy consumption in a two-user NOMA system, transmit power and computation offloading duration were optimized. Wang *et al.* [19] formulated a Stackelberg game for the NOMA-enabled, two-user MEC network, in which the user set acted as a leader while the MEC server acted as a follower, investigating total execution time minimization. A multiple-user MEC network in which the users offload their tasks using uplink NOMA to the edge cloud was considered in [20]–[22]. In [22], a weighted total energy consumption minimization problem was investigated in which the optimization variables were communication delay, uplink-downlink transmit power, and user task shares. Delay minimization for the NOMA-enabled multiple-user MEC network was investigated in [20], [21]. The optimization variables in [20] were transmission time and share of computation of each task that was to be computed. Fang *et al.* [21] improved the overall system delay compared to the proposed optimization strategy in [20] by considering transmit power allocation and share of computational resource allocation. The proposed solution was shown to be optimal under a negligible cloud computation delay assumption (*i.e.,* a very high edge cloud's computational resource). In practice, an MEC network may have limited edge cloud resources, and for such cases, the considered optimization problem is non-convex, and the proposed solution may not guarantee an optimal solution. Furthermore, computation resource allocation at the edge cloud and user devices was not investigated in these works [20]–[22]. Li *et al.* [23] investigated the problem of minimizing the total task completion energy for a multi-user multi-edge server network in which users offload their tasks via NOMA uplink to each edge cloud. Interference between transmission links to different edge clouds is avoided by bandwidth splitting.

Simultaneous offloading to a mobile peer and the edge cloud using NOMA has been investigated [24]. Compared to the non-cooperative cases and OMA, the proposed scheme has been shown to greatly reduce energy consumption. In [25], task offloading to multiple mobile peers using downlink NOMA was studied in case of unavailability of direct link from the computing user to the edge cloud. According to the proposed MEC frameworks in [24] or [25], a mobile peer acted as a relay to forward task data from the computing user to the edge cloud and helped by computing part of the computing user's task. However, no incentive scheme was proposed for the assisting mobile peer, and therefore, the application of this work in practice may be limited.

Different from all existing works, in this paper, we aim to solve the problem of balancing the communication and computation resources between two sets of users: a set of CUEs with computationally-intensive tasks and an underutilized uplink channel, and a set of DUEs, having high data rate demands and idle processors. In doing so, each CUE's task completion time is reduced and each DUE receives more information (compared to orthogonal transmission) from the BS. Compared to a monetary-incentive framework for edge computing networks [14], [15], our proposed framework has the advantage that along with efficiently computing the CUE's task, it also provides higher downlink rate gain to the DUEs in the network, and the network provider does not face issues that are associated with implementing monetary based incentives for helpers (*e.g.*, creating a secure framework for financial transactions in the network or considering a computing user's unwillingness to adhere to a certain monetary incentive). Since the DUEs' incentives are generated by efficiently allocating computation and communication resources in the network, the proposed framework can be easily implemented by a centralized controller. Furthermore, compared to the research works that consider task offloading only to the edge cloud (*i.e.*, where there is no requirement of developing an incentive framework) [16]–[22], the proposed strategy can use under-utilized computational resources available in the network more efficiently and can be more effective, particularly when CUEs have computationally-intensive tasks and there exists a set of DUEs, each with a high computational capability. The present work is based on our preliminary study in [1]. Compared to [1], the major improvements in this present study are as follows: (*i.*) In [1], we consider the pairing between a CUE and a DUE only when the CUE-to-DUE link has a better channel quality than the CUE-to-BS link. The uplink and downlink NOMA-enabled CUE-DUE pairing is not investigated when the CUE-to-DUE link is of lower channel quality than the CUE-to-BS link. In the journal version, we propose a separate transmission protocol for the CUE-DUE pairing scheme for each of these channel conditions. The proposed transmission protocol is improved as it enables interference-free transmission in the CUE-DUE and CUE-BS links under a certain channel condition, resulting in improvement in system performance. (*ii.*) Unlike the conference version, computational resource allocation at the user devices is performed to further improve performance of the proposed system model. (*iii.*) The dimensionality of the edge cloud resource allocation problem is reduced, and a more rigorous evaluation of the proposed system model is performed with many state-of-the-art approaches.

III. System Model and Problem Formulation

Consider a wireless BS associated with an edge cloud that provides computing facilities to a set $\mathcal{N} = \{1, 2, .., N\}$ of N CUEs. There is also a set $\mathcal{M} = \{1, 2, .., M\}$ of M DUEs that intend to receive information from the BS. We assume that the BS has already allocated an orthogonal uplink bandwidth to each CUE and an orthogonal downlink bandwidth to each DUE.[2] Let $g_{i,j}$, $g_{i,\mathrm{BS}}$, $g_{\mathrm{BS},i}$, and $g_{\mathrm{BS},j}$ be, respectively, the channel gain of CUE i to DUE j, CUE i to BS, BS to CUE i, and BS to DUE j links, where $i \in \mathcal{N}, j \in \mathcal{M}$. Each link experiences quasi-static Rayleigh fading. The computational capability of the user

[2]In [26], [27], bandwidth optimization strategies for MEC systems were studied. Such bandwidth allocation strategies can be employed along with our proposed framework.

$i \in \mathcal{N} \bigcup \mathcal{M}$ is denoted by $f_i^{\max}$ (in cycles/s). The task of each CUE i is given by $\phi_i = (\beta_i, b_i)$, where b_i indicates the task input data size, and β_i stands for the number of CPU cycles required to compute per bit of the task. The calculation method for these task parameters is described in [28]. Similar to [6], [7], [9], [10], we consider splittable tasks. Hence, each CUE can partially offload its task to other computing devices. Since the processors at the DUEs are idle, a DUE can compute a share of a CUE's computation tasks if it receives a proper incentive from the CUE. Each CUE has two choices:

- *Cloud-Only Mode:* In this case, the CUE offloads a share of computation bits to the edge cloud. Since computational resources from the DUEs are not utilized, the computing power applied to the task is less. However, communication resources available at the CUE are fully utilized for task offloading and thus the offloading delay is minimized.
- *Joint DUE-Cloud Offloading Mode:* The sum transmission rate from BS to CUE i and DUE j is higher in downlink NOMA than the OMA-based transmission to DUE j, when a CUE i has a higher downlink channel gain than a DUE j, *i.e.,* $g_{\text{BS},i} > g_{\text{BS},j}$ [29]. Utilizing this fact, we consider a strategy in which the BS sends information (intended for the DUE) to CUE i and DUE j in downlink NOMA and the CUE relays the information to the DUE, resulting in more information being received at the DUE and thus motivating the DUE to compute the CUE's task. Using uplink NOMA, CUE i sends the incentive bits to DUE j as well as shares of its computation task to DUE j and the edge cloud. Although compared to the cloud-only mode, offloading delay for CUE i's task may increase due to the CUE sending excess uplink information, overall a more efficient communication resource utilization is achieved due to enabling uplink and downlink NOMA. Also, more computing power is applied to the CUE's task.

For the latter mode, we assume that each DUE serves a maximum of one CUE and that each CUE offloads its task to a maximum of one DUE to reduce the system complexity. An illustration of a network with three CUEs and three DUEs in which CUEs 1 and 2 are in joint DUE-cloud offloading mode, and CUE 3 is in cloud-only mode is shown in Fig. 1. We now describe the joint DUE-cloud offloading mode for the following channel conditions: (1.) $g_{i,j} > g_{i,\text{BS}}$, and (2.) $g_{i,j} < g_{i,\text{BS}}$. Then, we describe the cloud-only mode.

A. Joint DUE-Cloud Offloading for Channel Condition 1

Consider a CUE $i \in \mathcal{M}$ that is paired with a DUE $j \in \mathcal{N}$ to operate in the joint DUE-cloud offloading mode. In the uplink direction, the CUE offloads a portion of its task to the DUE and the edge cloud and forwards incentive bits to the DUE in two time slots. Let $t_{i,j}^1$ and $t_{i,j}^2$ be the duration of time slots 1 and 2, respectively. In the first time slot, $t_{i,j}^1$, the CUE offloads $b_{i,j}$ and $b_{i,j}^{\text{EC},1}$ bits of its task to DUE j and the edge cloud, respectively, using NOMA. The BS sends information bits (intended for the DUE) to the CUE and the DUE within the time duration $t_{i,j}^1 + t_{i,j}^2$ using downlink NOMA. Let $b_{i,j}^r$ denote the number of bits that the BS sends to the CUE within the time duration $t_{i,j}^1 + t_{i,j}^2$.

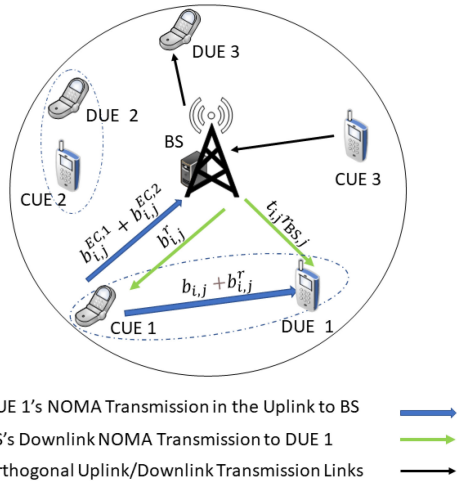


Fig. 1. An illustration of network with three CUEs and three DUEs.

In the second time slot, the CUE sends these bits to DUE j and offloads $b_{i,j}^{\text{EC},2}$ bits of its task to the BS using NOMA. Also, the CUE computes $b_{i,j}^l$ bits of its task ϕ_i locally at its own processor. Therefore, the relationship between local computation bits and offloaded bits to the edge cloud and DUE j can be expressed as $b_{i,j}^l + b_{i,j}^{\text{EC},1} + b_{i,j}^{\text{EC},2} + b_{i,j} = b_i$. In case of the mobile edge computing network shown in Fig. 1, the CUE 1 and DUE 1 is operating in this mode. Fig. 2(a) shows the operations at different devices for CUE's task completion, and task offloading data flow when a CUE i is paired with DUE j to operate in this mode. Fig. 2(b) shows uplink-downlink transmission protocol. We now characterize the delay and energy for the CUE's task computation and also quantify the DUE's incentive.

1) CUE's NOMA Transmission: Let $P_{i,j}^1$ and $P_{i,j}^{\text{BS},1}$ indicate the transmit power for CUE i to DUE j and CUE i to BS links, respectively, at the first time slot. Also, let B_u be the bandwidth of the uplink channel allocated to CUE i. DUE j utilizes SIC (since $g_{i,j} > g_{i,BS}$) to detect the signal intended for it after decoding the signal intended for the BS. The BS detects its signal by regarding the signal intended for DUE j as interference. Therefore, the transmission rate (in b/s) from CUE i to DUE j and to the BS at the first time slot is given by [30]:

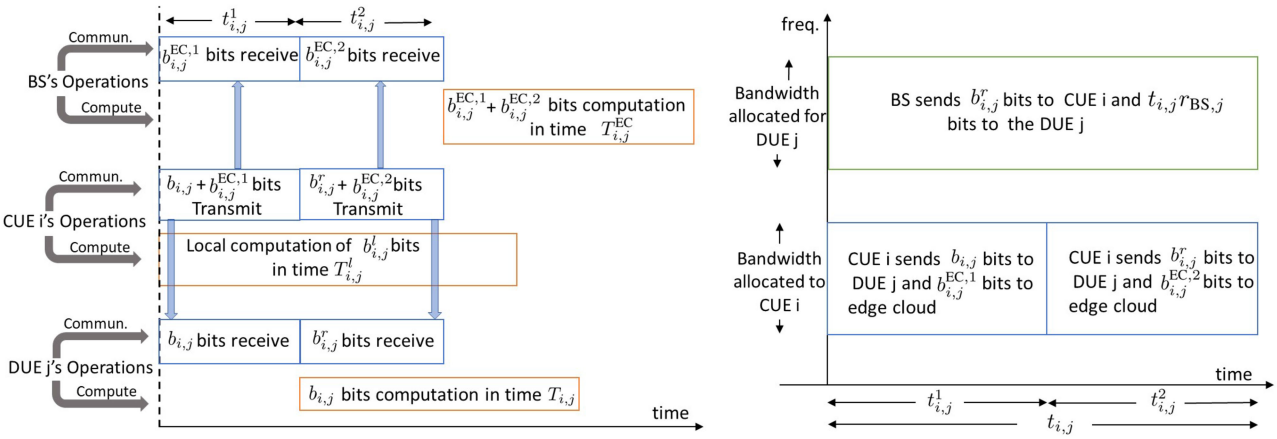
$$r_{i,j}^1 = B_u \log\left(1 + \frac{P_{i,j}^1 g_{i,j}}{N_0}\right), \text{ and} \tag{1}$$

$$r_{i,j}^{\text{BS},1} = B_u \log\left(1 + \frac{P_{i,j}^{\text{BS},1} g_{i,\text{BS}}}{N_0 + P_{i,j}^1 g_{i,\text{BS}}}\right), \tag{2}$$

respectively. Here, N_0 is the noise power. Then, we have:

$$P_{i,j}^1 = \frac{N_0}{g_{i,j}} f\left(\frac{b_{i,j}}{t_{i,j}^1 B_u}\right), \text{ and} \tag{3}$$

$$P_{i,j}^{\text{BS},1} = N_0 \left(\left(\frac{1}{g_{i,\text{BS}}} - \frac{1}{g_{i,j}}\right) f\left(\frac{b_{i,j}^{\text{EC},1}}{t_{i,j}^1 B_u}\right) + \frac{1}{g_{i,j}} f\left(\frac{b_{i,j} + b_{i,j}^{\text{EC},1}}{t_{i,j}^1 B_u}\right) - \frac{1}{g_{i,j}} f\left(\frac{b_{i,j}}{t_{i,j}^1 B_u}\right) \right), \tag{4}$$



(a) (b)

Fig. 2. Joint DUE-cloud Offloading for $g_{i,j} > g_{i,\mathrm{BS}}$: (a) Operations at different devices for CUE i's task completion, and (b) Uplink and downlink transmission.

where $f(x) = 2^x - 1$. Here, (3) and (4) are obtained using (1), (2), and the relationships $b_{i,j} = t^1_{i,j} r_{i,j}$, $b^{\mathrm{EC},1}_{i,j} = t^1_{i,j} r^{\mathrm{BS},1}_{i,j}$.

We use $P^2_{i,j}$ and $P^{\mathrm{BS},2}_{i,j}$ to denote CUE i's transmit power to DUE j and BS, respectively, at the second time slot. Using SIC, the transmission rate $r^2_{i,j}$ from CUE i to DUE j during the second time slot is:

$$r^2_{i,j} = B_u \log\left(1 + \frac{P^2_{i,j} g_{i,j}}{N_0}\right). \tag{5}$$

Since, the DUE receives $b^r_{i,j}$ bits within the duration of $t^2_{i,j}$, we have:

$$P^2_{i,j} = \frac{t^2_{i,j} N_0}{g_{i,j}} f\left(\frac{b^r_{i,j}}{t^2_{i,j} B_u}\right). \tag{6}$$

Along with the signal intended for it, the BS also receives the signal intended to the DUE as an interfering signal. Since the BS is the original source for this interfering signal intended for the DUE,[3] it can perfectly decode this signal and eliminate the interference. Therefore, the achievable transmission rate $r^{\mathrm{BS},2}_{i,j}$ from CUE i to the BS during the second time slot is:

$$r^{\mathrm{BS},2}_{i,j} = B_u \log\left(1 + \frac{P^{\mathrm{BS},2}_{i,j} g_{i,\mathrm{BS}}}{N_0}\right). \tag{7}$$

Using (7) and the relationship $b^{\mathrm{EC},2}_{i,j} = t^2_{i,j} r^{\mathrm{BS},2}_{i,j}$, we have:

$$P^{\mathrm{BS},2}_{i,j} = \frac{t^2_{i,j} N_0}{g_{i,\mathrm{BS}}} f\left(\frac{b^{\mathrm{EC},2}_{i,j}}{t^2_{i,j} B_u}\right). \tag{8}$$

2) Overall Delay and Energy Analysis for CUEs: Let $T^l_{i,j}$, $T_{i,j}$, and $T^{\mathrm{EC}}_{i,j}$ be the computation time of the share of ϕ_i at CUE i, DUE j, and the edge cloud, respectively. The local computation delay can be expressed as:

$$T^l_{i,j} = \frac{\beta_i b^l_{i,j}}{f_i}, \tag{9}$$

[3]The signal was sent earlier by the BS.

where f_i is the allocated CPU power (in cycles per second) at CUE i to compute $b^l_{i,j}$ bits and is upper bounded by the maximum frequency constraint $f^{\max}_i$, *i.e.*, $f_i \le f^{\max}_i$. CUE i's computational power consumption can be expressed as $p^l_i = \gamma_c f^3_i$, where γ_c denotes a constant related to the processor hardware architecture [10]. The computation energy at CUE i to compute $b^l_{i,j}$ bits can be expressed by:

$$E^l_{i,j} = p^l_i T^l_{i,j} = \gamma_c f^3_i T^l_{i,j} \overset{(a)}{=} \frac{\gamma_c \beta^3_i {b^l_{i,j}}^3}{{T^l_{i,j}}^2}, \tag{10}$$

where step (a) follows by replacing for f_i from (9). Similarly, the computation delay, $T_{i,j}$, at DUE j can be expressed as:

$$T_{i,j} = \frac{\beta_i b_{i,j}}{f_j}, \tag{11}$$

where f_j is the computation power allocated at DUE j to compute $b_{i,j}$ bits of the task and is also subject to the maximum frequency constraint $f^{\max}_j$, *i.e.*, $f_j \le f^{\max}_j$. Similar to the steps of (10), the computation energy at DUE j to compute $b_{i,j}$ bits can be expressed by:

$$E_{i,j} = \frac{\gamma_c \beta^3_i b^3_{i,j}}{T^2_{i,j}}. \tag{12}$$

Let F_i be the share of the edge cloud's computation power allocated to compute the task bits that CUE i offloads to the edge cloud. The total processing power at the edge cloud is F, *i.e.*, $\sum_{i=1}^{N} F_i \le F$. Then, the computation time at the edge cloud is:

$$T^{\mathrm{EC}}_{i,j} = \frac{\beta_i \left(b^{\mathrm{EC},1}_{i,j} + b^{\mathrm{EC},2}_{i,j}\right)}{F_i}. \tag{13}$$

The task completion times at DUE j and at the edge cloud are $t^1_{i,j} + T_{i,j}$ and $t^1_{i,j} + t^2_{i,j} + T^{\mathrm{EC}}_{i,j}$, respectively. Therefore, completion time of CUE i's task can be expressed as:

$$\mathcal{T}_{i,j} = \max\left(T^l_{i,j}, t^1_{i,j} + T_{i,j}, t^1_{i,j} + t^2_{i,j} + T^{\mathrm{EC}}_{i,j}\right). \tag{14}$$

The transmission delay in sending back the computation results is ignored since the size of the computation results is usually much smaller compared to that of the input data [6]–[10].

Using (3), (4), (6), (8), and (10), the total energy consumption at CUE i can be expressed as:

$$\mathcal{E}_{i,j} = t^1_{i,j} N_0 \Bigg(\left(\frac{1}{g_{i,\mathrm{BS}}} - \frac{1}{g_{i,j}} \right) f\left(\frac{b^{\mathrm{EC},1}_{i,j}}{t^1_{i,j} B_u} \right) + \frac{1}{g_{i,j}} f\left(\frac{b_{i,j} + b^{\mathrm{EC},1}_{i,j}}{t^1_{i,j} B_u} \right) \Bigg) + t^2_{i,j} N_0 \Bigg(\frac{1}{g_{i,j}} f\left(\frac{b^r_{i,j}}{t^2_{i,j} B_u} \right) + \frac{1}{g_{i,\mathrm{BS}}} f\left(\frac{b^{\mathrm{EC},2}_{i,j}}{t^2_{i,j} B_u} \right) \Bigg) + \frac{\gamma_c \beta_i^3 {b^l_{i,j}}^3}{{T^l_{i,j}}^2}. \quad (15)$$

3) BS's NOMA Transmission: The downlink NOMA transmission rate for BS to CUE i and BS to DUE j links are:

$$r_{\mathrm{BS},i} = B_d \log\left(1 + \frac{\alpha P_{\mathrm{BS}} g_{\mathrm{BS},i}}{N_0}\right), \quad (16)$$

and,

$$r_{\mathrm{BS},j} = B_d \log\left(1 + \frac{(1-\alpha) P_{\mathrm{BS}} g_{\mathrm{BS},j}}{N_0 + \alpha P_{\mathrm{BS}} g_{\mathrm{BS},j}}\right), \quad (17)$$

respectively, in case, $g_{\mathrm{BS},i} > g_{\mathrm{BS},j}$. Here, B_d is the bandwidth of the downlink channel allocated to DUE j, and $\alpha \in (0,1)$ is the power allocation factor for downlink transmission. The BS sends $b^r_{i,j}$ bits to CUE i within the downlink NOMA transmission duration of $t^1_{i,j} + t^2_{i,j}$. Hence, using (16), we have:

$$\alpha = \frac{N_0}{P_{\mathrm{BS}} g_{\mathrm{BS},i}} f\left(\frac{b^r_{i,j}}{\left(t^1_{i,j} + t^2_{i,j}\right) B_d} \right). \quad (18)$$

Substituting (18) into (17), we have:

$$r_{\mathrm{BS},j} = B_d \log\left(\frac{N_0 g_{\mathrm{BS},i} + P_{\mathrm{BS}} g_{\mathrm{BS},i} g_{\mathrm{BS},j}}{N_0 g_{\mathrm{BS},i} + N_0 g_{\mathrm{BS},j} f\left(\frac{b^r_{i,j}}{(t^1_{i,j} + t^2_{i,j}) B_d} \right)} \right). \quad (19)$$

4) DUE's Incentive Design: DUE j receives an increased number of bits in the downlink from the BS (compared to orthogonal transmission from BS to the DUE) within the computation offloading duration, which serves as an incentive to compute the task of CUE i. Specifically, to spend the energy $E_{i,j}$ in computing the task for CUE i, the incentive bit gain for DUE j is $b^r_{i,j} + (t^1_{i,j} + t^2_{i,j})(r_{\mathrm{BS},j} - r^{\mathrm{OMA}}_{\mathrm{BS},j})$ where $r^{\mathrm{OMA}}_{\mathrm{BS},j}$ is the orthogonal downlink achievable rate for BS to DUE j link, *i.e.*

$$r^{\mathrm{OMA}}_{\mathrm{BS},j} = B_d \log\left(1 + \frac{P_{\mathrm{BS}} g_{\mathrm{BS},j}}{N_0 g_{\mathrm{BS},j}}\right). \quad (20)$$

We define the utility, $U_{i,j}$, of DUE j when cooperating with CUE i, as:

$$U_{i,j} = b^r_{i,j} + \left(t^1_{i,j} + t^2_{i,j}\right)\left(r_{\mathrm{BS},j} - r^{\mathrm{OMA}}_{\mathrm{BS},j}\right) - \mathsf{k}_j E_{i,j}, \quad (21)$$

Where $\mathsf{k}_j > 0$ is the incentive needed in terms of the number of excess bits at the expense of per unit of computing energy at DUE j. We refer to k_j as the trading factor. DUE j decides to participate in the joint DUE-cloud offloading mode if $U_{i,j} \geq 0$.

Furthermore, the participating DUE j can impose a minimum computation frequency allocation $f^{\min}_j \in (0, f^{\max}_j)$. If such a restriction is imposed, using (12), (21), and the constraint $U_{i,j} \geq 0$, we have: Incentive bit gain$\geq \mathsf{k}_j \gamma_c \beta_i {f^{\min}_j}^2 b_{i,j}$. Therefore, by imposing the minimum computation frequency allocation constraint, the DUE has the flexibility to receive compensation for the number of bits that it computes for the CUE in terms of the incentive bit gain that it receives for a fixed value of k_j, γ_c, and β_i. When a DUE is willing to participate in the joint-DUE cloud offloading mode based on the incentive bit gain as compensation for its computation energy consumption, it can declare $f^{\min}_j = 0$, and if the DUE decides to participate based on the incentive bit gain as completely dependent on the number of bits that it computes, it can declare $f^{\min}_j = f^{\max}_j$. Each DUE j may choose the value of the trading factor based on excess information it would like to receive in the downlink and its remaining battery energy. The effect of various values of the trading factor is discussed in Section VI.

Remark 1: If $g_{\mathrm{BS},i} < g_{\mathrm{BS},j}$, downlink NOMA transmission rate for DUE j is upper bounded by $r^{\mathrm{OMA}}_{\mathrm{BS},j}$. Thus, DUE j can not receive any incentive bit gain by participating with CUE i in joint DUE-cloud offloading mode. Hence, the CUE operates in cloud-only mode, which we discuss in Section III-C.

B. Joint DUE-Cloud Offloading for Channel Condition 2

We now briefly describe the joint DUE-Cloud offloading if the channel condition $g_{i,j} < g_{i,\mathrm{BS}}$ is satisfied. Note that Remark 1 also holds in this case. Let $t_{i,j}$ be the transmission duration. The CUE offloads $b^{\mathrm{EC}}_{i,j}$ bits of its task ϕ_i to the edge cloud and sends $b_{i,j} + b^r_{i,j}$ bits to DUE j in the time of $t_{i,j}$, where $b_{i,j}$ is the number of bits of CUE i's task to be computed at DUE j, $b^r_{i,j}$ represents the number of incentive bits sent by the BS to the CUE in time of $t_{i,j}$ using downlink NOMA. Furthermore, $b^l_{i,j} = b_i - b^{\mathrm{EC}}_{i,j} - b_{i,j}$ bits of the task ϕ_i is computed at CUE i. The operation at different devices for CUE's task completion and task data flow is shown in Fig. 3(a). The uplink-downlink transmission protocol is shown in Fig. 3(b). Since we have $g_{i,j} < g_{i,\mathrm{BS}}$, using SIC at the BS, the transmission rate from CUE i to BS is $r^{\mathrm{BS}}_{i,j} = B_u \log(1 + P^{\mathrm{BS}}_{i,j} g_{i,\mathrm{BS}}/N_0)$ and rate $r_{i,j}$ from CUE i to DUE j is $r_{i,j} = B_u \log(1 + P_{i,j} g_{i,j}/(N_0 + P^{\mathrm{BS}}_{i,j} g_{i,j}))$. The overall energy consumption at CUE i to complete the task ϕ_i is:

$$\mathcal{E}'_{i,j} = t_{i,j} N_0 \Bigg(\left(\frac{1}{g_{i,j}} - \frac{1}{g_{i,\mathrm{BS}}} \right) f\left(\frac{b_{i,j} + b^r_{i,j}}{t_{i,j} B_u} \right) + \frac{1}{g_{i,\mathrm{BS}}} f\left(\frac{b_{i,j} + b^{\mathrm{EC}}_{i,j} + b^r_{i,j}}{t_{i,j} B_u} \right) \Bigg) + \frac{\gamma_c \beta_i^3 {b^l_{i,j}}^3}{{T^l_{i,j}}^2}, \quad (22)$$

and CUE i's task completion time is:

$$\mathcal{T}'_{i,j} = \max\left(T^l_{i,j}, t_{i,j} + T_{i,j}, t_{i,j} + T^{\mathrm{EC}}_{i,j}\right). \quad (23)$$

Here, $T^l_{i,j}$, and $T_{i,j}$ are defined in (9) and (11), respectively, and $T^{\mathrm{EC}}_{i,j} = \beta_i b^{\mathrm{EC}}_{i,j}/F_i$. The utility, $U'_{i,j}$, of DUE j can be expressed as:

$$U'_{i,j} = b^r_{i,j} + t_{i,j}(r'_{\mathrm{BS},j} - r^{\mathrm{OMA}}_{\mathrm{BS},j}) - \mathsf{k}_j E_{i,j}, \quad (24)$$

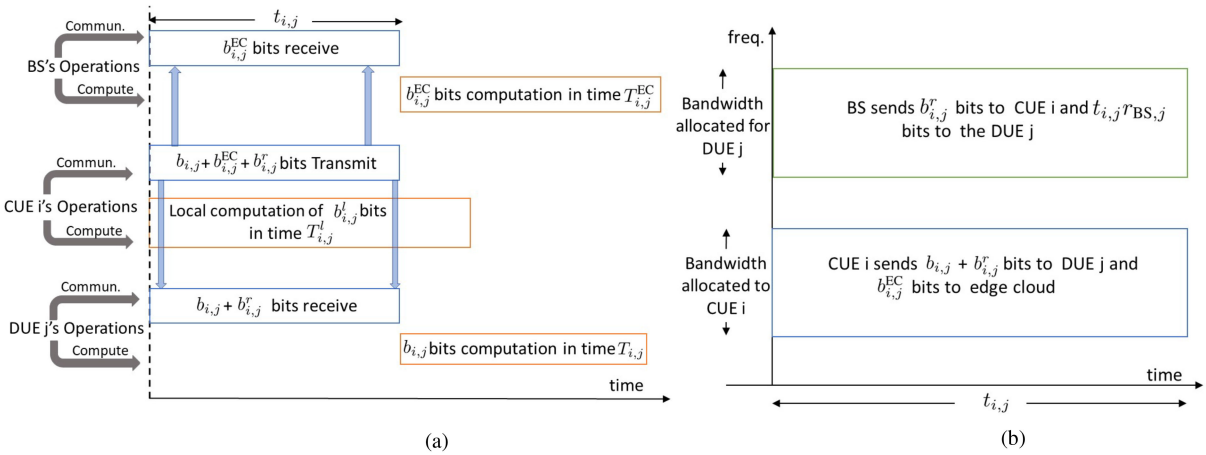


Fig. 3. Joint DUE-cloud Offloading for $g_{i,j} < g_{i,\mathrm{BS}}$ (a) Operations at different devices for CUE i's task completion and task data flow (b) Uplink and downlink transmission.

where $E_{i,j}$, and $r_{\mathrm{BS},j}^{\mathrm{OMA}}$ are expressed in (12) and (20), respectively and

$$r'_{\mathrm{BS},j} = B_d \log\left(\frac{N_0 g_{\mathrm{BS},i} + P_{\mathrm{BS}} g_{\mathrm{BS},i} g_{\mathrm{BS},j}}{N_0 g_{\mathrm{BS},i} + N_0 g_{\mathrm{BS},j} f\left(\frac{b_{i,j}^r}{t_{i,j} B_d}\right)}\right).$$

Remark 2: If the CUE simultaneously transmits $b_{i,j}^{\mathrm{EC},1} + b_{i,j}^{\mathrm{EC},2}$ and $b_{i,j}^r + b_{i,j}$ bits (intended for the BS and DUE j, respectively) within one time slot when $g_{i,j} > g_{i,\mathrm{BS}}$, the interference cannot be eliminated at the BS since a superimposed signal of task bits and relay bits are received at the BS. The proposed transmission protocol in Section III-A is efficient in the sense that interference at both DUE j and the BS can be eliminated in slot 2. However, interference cancellation at both CUE-DUE and CUE-BS links is not possible when $g_{i,j} < g_{i,\mathrm{BS}}$.

C. Cloud-Only Offloading

In this case, CUE k's ($k \in \{1, .., N\}$) task completion time is given by:

$$\mathcal{T}_k = \max\left(T_k^l, t_k + T_k^{\mathrm{EC}}\right). \tag{25}$$

Here, t_k, $T_k^{\mathrm{EC}} = \beta_k b_k^{\mathrm{EC}}/F_k$, and $T_k^l = \beta_k\, b_k^l/f_k$ are, respectively, delay to offload b_k^{EC} bits, edge cloud's computation time of b_k^{EC} bits, and CUE k's computation time of b_k^l bits. Hence, we have $b_k^l + b_k^{\mathrm{EC}} = b_k$.

The communication energy consumption to offload b_k^{EC} bits to the edge cloud in time t_k is $t_k N_0/g_{k,\mathrm{BS}} f(b_k^{\mathrm{EC}}/t_k B_u)$, and the energy consumption to compute b_k^l bits locally in time T_k^l is $\gamma_c \beta_k^3 {b_k^l}^3/{T_k^l}^2$. Therefore, the total energy consumption at CUE k to complete the task ϕ_k is:

$$\mathcal{E}_k = \frac{t_k N_0}{g_{k,\mathrm{BS}}} f\left(\frac{b_k^{\mathrm{EC}}}{t_k B_u}\right) + \frac{\gamma_c \beta_k^3 {b_k^l}^3}{{T_k^l}^2}. \tag{26}$$

If a DUE $j \in \mathcal{M}$ is not assigned any CUE to operate in the joint DUE-cloud offloading mode, it receives information at the downlink orthogonal transmission rate, *i.e.*, at $r_{\mathrm{BS},j}^{\mathrm{OMA}}$.

IV. Problem Formulation

In the considered network, joint DUE-cloud offloading may be more beneficial (in terms of task completion time) than the cloud-only mode for one or more CUEs. Therefore, to minimize the task completion time of all CUEs, each of these CUEs should be paired with a DUE, while the rest of the CUEs should employ cloud-only mode. To formulate the assignment decision for each CUE and DUE in the network, we denote the following sets: Let π denote a set partition of $\mathcal{N} \cup \mathcal{M}$ such that there is a CUE and a maximum of one DUE in each subset, and let Π denotes the set of all these possible partitions. For instance, with $\mathcal{N} = \{1, 2\}$ and $\mathcal{M} = \{1\}$, we have:

$$\Pi = \{\{\{1,1\},\{2\}\}, \{\{1\},\{2,1\}\}, \{\{1\},\{2\}\}\}. \tag{27}$$

In each subset, the first and second members are, respectively, the CUE and DUE. For instance, the partition $\{\{1,1\},\{2\}\}$ states that CUE 1 is paired with DUE 1 to operate in joint DUE-cloud offloading mode, and CUE 2 is in cloud-only mode. We refer to each of these partitions as a CUE-DUE assignment. Let ρ_π and ζ_π indicate the group of all subsets of π with size one and two, respectively. For example, if $\pi = \{\{1\},\{2,1\}\}$, ζ_π consists of $\{2,1\}$ and ρ_π includes $\{1\}$. Also, let ζ_π^1, ζ_π^2 ($\zeta_\pi^1 \bigcup \zeta_\pi^2 = \zeta_\pi$) be the group of all subsets of π such that for each subset $\{i,j\} \in \zeta_\pi^1$ and $\{m,n\} \in \zeta_\pi^2$, we have $g_{i,j} > g_{i,\mathrm{BS}}$ and $g_{m,n} < g_{m,\mathrm{BS}}$, respectively. For example, in case $\pi = \{\{1\},\{2,1\}\}$ and assuming $g_{2,1} > g_{2,\mathrm{BS}}$, we have $\zeta_\pi^1 = \{2,1\}$, and $\zeta_\pi^2 = \emptyset$.

We define the maximum task completion time among all the CUEs as follows:

$$\mathcal{T}^N = \max\left(\max_{\{i,j\}\in\zeta_\pi^1} \mathcal{T}_{i,j}, \max_{\{m,n\}\in\zeta_\pi^2} \mathcal{T}_{m,n}, \max_{k\in\rho_\pi} \mathcal{T}_k\right). \tag{28}$$

In this paper, we aim to minimize $\mathcal{T}^N$ considering the following optimization variables: Deciding the CUEs that employ cloud-only mode, assigning a DUE to each CUE that employs joint DUE-cloud offloading mode, offloaded computation shares, the number of incentive bits that each CUE relays to the paired DUE, transmission time, the edge cloud resource distribution among

the CUEs, and computation resource (equivalently computation time) at each user device. Formally, the problem is:

$$\min_{\pi\in\Pi,\boldsymbol{F},\boldsymbol{b_\pi},\boldsymbol{t_\pi},\boldsymbol{T_\pi}} \quad \mathcal{T}^N$$

$$\text{s.t.} \quad U_{i,j}, U'_{m,n} \geq 0,\ \forall\{i,j\}\in\zeta_\pi^1, \{m,n\}\in\zeta_\pi^2 \tag{29a}$$

$$\mathcal{E}_{i,j} \leq \mathcal{E}_{\text{th},i},\ \ \mathcal{E}'_{m,n} \leq \mathcal{E}_{\text{th},m},\ \ \mathcal{E}_k \leq \mathcal{E}_{\text{th},k}$$

$$\forall\{i,j\}\in\zeta_\pi^1, \{m,n\}\in\zeta_\pi^2, k\in\rho_\pi \tag{29b}$$

$$\sum_{l=1}^{N} F_l \leq F \tag{29c}$$

$$\frac{\beta_i b_{i,j}^l}{f_i^{\max}} \leq T_{i,j}^l, \quad \forall\{i,j\}\in\zeta_\pi \tag{29d}$$

$$\frac{\beta_i b_{i,j}}{f_j^{\max}} \leq T_{i,j}, \quad \forall\{i,j\}\in\zeta_\pi \tag{29e}$$

$$\frac{\beta_i b_{i,j}}{f_j^{\min}} \geq T_{i,j}, \quad \forall\{i,j\}\in\zeta_\pi \tag{29f}$$

$$\frac{\beta_k b_k^l}{f_k^{\max}} \leq T_k^l, \quad \forall k\in\rho_\pi \tag{29g}$$

$$b_{i,j}^l + b_{i,j} + b_{i,j}^{\text{EC},1} + b_{i,j}^{\text{EC},2} = b_i,\ \{i,j\}\in\zeta_\pi^1 \tag{29h}$$

$$b_{m,n}^l + b_{m,n} + b_{m,n}^{\text{EC}} = b_i,\ \{m,n\}\in\zeta_\pi^2 \tag{29i}$$

$$b_k^l + b_k^{\text{EC}} = b_k \quad \forall k\in\rho_\pi. \tag{29j}$$

Here, $\boldsymbol{b_\pi}$ is the set consisting of variables $b_{i,j}^l$, $b_{i,j}$, $b_{i,j}^{\text{EC},1}$, $b_{i,j}^{\text{EC},2}$, $b_{i,j}^r$, $b_{m,n}^l$, $b_{m,n}$ $b_{m,n}^{\text{EC}}$, $b_{m,n}^r$, b_k^l and b_k^{EC}, and $\boldsymbol{t_\pi}$ is the set consisting of variables $t_{i,j}^1, t_{i,j}^2, t_{m,n}$ and t_k for $\{i,j\}\in\zeta_\pi^1$, $\{m,n\}\in\zeta_\pi^2, k\in\rho_\pi$. Also, $\boldsymbol{T_\pi}$ is the set consisting of variables $T_{i,j}^l$, $T_{i,j}$ and T_k^l for $\{i,j\}\in\zeta_\pi$, $k\in\rho_\pi$, and $\boldsymbol{F}$ is the set consisting of variables F_l, for $l\in\mathcal{N}$. The constraints (29a) capture DUEs' incentive requirements. The constraints (29b) indicate that each CUE i's energy consumption is bounded by an energy threshold. Constraint (29c) specifies that the computing power allocated by the edge cloud to all CUEs cannot exceed its total computing power. The constraints (29d), (29e), and (29g) guarantee that the allocated computation frequencies of the CUEs and DUEs stay below their respective limits. DUE j's minimum computation resource allocation is captured in (29f). Finally, (29h), (29i), and (29j) guarantee that the total task bits for each CUE is equal to the sum of local computation bits and the offloaded bits. The difficulty in solving (29) can be explained as follows: (*i.*) Given a CUE-DUE assignment $\pi\in\Pi$, (29) is non-convex, (*ii.*) An exhaustive search over all possible assignments is needed to solve the CUE-DUE assignment problem. Next, we present a sub-optimal, low-complexity solution that can be implemented in a centralized manner. We assume that knowledge of the network state, *i.e.*, the information on CUEs' computation task parameters and global channel state is available at a central controller. Therefore, the controller can design and coordinate the communication and computation cooperation among the CUEs and DUEs. This serves as a performance upper bound (or completion time lower bound) for practical situations where only partial knowledge of the network state is available.

Remark 3: The utility function is designed such that if $b_{i,j}=0$, $\{i,j\}\in\zeta_\pi^1$, the constraint (29a) is satisfied at equality (*i.e.*, $U_{i,j}=0$) by setting $b_{i,j}^r=0$, which signifies that if CUE i does not offload any computation bits to DUE j, it does not need to forward any information to DUE j from the BS and the DUE receives information according to the rate $r_{\text{BS},j}^{\text{OMA}}$. Similar observations can be drawn for each CUE-DUE pair $\{m,n\}\in\zeta_\pi^2$.

V. Proposed Solution

The problem (29) is divided into two sub-problems as follows: *A.* computation time allocation, transmission time allocation, sharing of task bits and incentive bits, and CUE-DUE assignment optimization, and *B.* cloud computation resource allocation. The sub-problems A and B are solved sequentially at each iteration, and this procedure is repeated until the completion time objective converges. Let 'p' denote the iteration counter of the proposed algorithm.

A. Allocation of Computation Time, Transmission Time, Sharing of Task and Incentive Bits, and CUE-DUE Assignment Optimization

In each iteration p, (29) is solved for a fixed cloud resource allocation $\boldsymbol{F}^p=[F_1^p,..,F_N^p]$, where F_i^p is the cloud computing power assigned to CUE i, $i\in\{1,..,N\}$, at iteration p. Consequently, we have the following optimization problem:

$$\min_{\pi\in\Pi,\boldsymbol{b_\pi},\boldsymbol{t_\pi},\boldsymbol{T_\pi}} \max\left(\max_{\{i,j\}\in\zeta_\pi^1}\mathcal{T}_{i,j}, \max_{\{m,n\}\in\zeta_\pi^2}\mathcal{T}'_{m,n}, \max_{k\in\rho_\pi}\mathcal{T}_k\right)$$
$$\text{s.t.} \quad \text{(29a), (29b), (29d)–(29j)} \tag{30}$$

We first discuss the solution strategy of (30) for a fixed CUE-DUE assignment π. For a fixed CUE-DUE assignment π, (30) decouples into the following sub-problems:

$$\min_{\substack{t_{i,j},T_{i,j}^l,T_{i,j},b_{i,j}^l\\ b_{i,j},b_{i,j}^{\text{EC},1},b_{i,j}^{\text{EC},2},b_{i,j}^r}} \quad \mathcal{T}_{i,j}$$

$$\text{s.t.} \quad U_{i,j}\geq 0$$
$$\mathcal{E}_{i,j}\leq\mathcal{E}_{\text{th},i}$$
$$\frac{\beta_i b_{i,j}^l}{f_i^{\max}}\leq T_{i,j}^l,$$
$$\frac{\beta_i b_{i,j}}{f_j^{\max}}\leq T_{i,j},$$
$$\frac{\beta_i b_{i,j}}{f_j^{\min}}\geq T_{i,j},$$
$$b_{i,j}^l + b_{i,j} + b_{i,j}^{\text{EC},1} + b_{i,j}^{\text{EC},2} = b_i \tag{31}$$

for all $\{i,j\} \in \zeta_\pi^1$:

$$\begin{aligned}
\min_{\substack{t_{m,n},T^l_{m,n},T_{m,n},b^l_{m,n}\\ b_{m,n},b^{\text{EC}}_{m,n},b^r_{m,n}}} \quad & \mathcal{T}'_{m,n} \\
\text{s.t.} \quad & U'_{m,n} \geq 0 \\
& \mathcal{E}'_{m,n} \leq \mathcal{E}_{\text{th},m} \\
& \frac{\beta_m b^l_{m,n}}{f^{\max}_n} \leq T^l_{m,n}, \\
& \frac{\beta_m b_{m,n}}{f^{\max}_n} \leq T_{m,n}, \\
& \frac{\beta_n b_{m,n}}{f^{\min}_n} \geq T_{m,n}, \\
& b^l_{m,n} + b_{m,n} + b^{\text{EC}}_{m,n} = b_m \qquad (32)
\end{aligned}$$

for all $\{m,n\} \in \zeta_\pi^2$, and:

$$\begin{aligned}
\min_{t_k, b^l_k b^c_{k,\text{EC}}} \quad & \mathcal{T}_k \\
\text{s.t.} \quad & \mathcal{E}_k \leq \mathcal{E}_{\text{th},k} \\
& \frac{\beta_k b^l_k}{f^{\max}_k} \leq T^l_k, \\
& b^l_k + b^{\text{EC}}_k = b_k \qquad (33)
\end{aligned}$$

for all $k \in \rho_\pi$. Next, we provide a solution strategy for each of these sub-problems. We leverage these solutions to solve (30).

1) Optimal Computation Time, Transmission Time, Sharing of Task and Incentive Bits: Using (12)–(15), (19), and (21), the optimization problem in (31) can be written as:

$$\begin{aligned}
\min_{\boldsymbol{\chi}} \quad & V \\
\text{s.t.} \quad & T^l_{i,j} \leq V \\
& t^1_{i,j} + T_{i,j} \leq V \\
& t^1_{i,j} + t^2_{i,j} + \frac{\beta_i (b^{\text{EC},1}_{i,j} + b^{\text{EC},2}_{i,j})}{F^p_i} \leq V \\
& (t^1_{i,j} + t^2_{i,j}) B_d \log\left(\frac{N_0 g_{\text{BS},i} + P_{\text{BS}} g_{\text{BS},i} g_{\text{BS},j}}{N_0 g_{\text{BS},i} + N_0 g_{\text{BS},j} f\left(\frac{b^r_{i,j}}{(t^1_{i,j}+t^2_{i,j})B_d}\right)} \right) \\
& + b^r_{i,j} - (t^1_{i,j} + t^2_{i,j}) r^{\text{OMA}}_{\text{BS},j} - \mathsf{k}_j \frac{\gamma_c \beta_i^3 b^3_{i,j}}{{T_{i,j}}^2} \geq 0 \\
& t^1_{i,j} N_0 \left(\left(\frac{1}{g_{i,\text{BS}}} - \frac{1}{g_{i,j}} \right) f\left(\frac{b^{\text{EC},1}_{i,j}}{t^1_{i,j} B_u} \right) \right. \\
& \left. + \frac{1}{g_{i,j}} f\left(\frac{b_{i,j} + b^{\text{EC},1}_{i,j}}{t^1_{i,j} B_u} \right) \right) + t^2_{i,j} N_0 \left(\frac{1}{g_{i,j}} f\left(\frac{b^r_{i,j}}{t^2_{i,j} B_u} \right) \right. \\
& \left. + \frac{1}{g_{i,\text{BS}}} f\left(\frac{b^{\text{EC},2}_{i,j}}{t^2_{i,j} B_u} \right) \right) + \frac{\gamma_c \beta_i^3 {b^l_{i,j}}^3}{{T^l_{i,j}}^2} \leq \mathcal{E}_{\text{th},i} \\
& \frac{\beta_i b^l_{i,j}}{f^{\max}_i} \leq T^l_{i,j}, \quad \frac{\beta_i b_{i,j}}{f^{\max}_j} \leq T_{i,j} \quad \frac{\beta_i b_{i,j}}{f^{\min}_j} \geq T_{i,j} \\
& b^l_{i,j} + b_{i,j} + b^{\text{EC},1}_{i,j} + b^{\text{EC},2}_{i,j} = b_i \qquad (34)
\end{aligned}$$

Here, V is a slack variable and $\boldsymbol{\chi}$ is the set containing variables V, $t_{i,j}$, $T^l_{i,j}$, $T_{i,j}$, $b^l_{i,j}$ $b_{i,j}$, $b^{\text{EC},1}_{i,j}$, $b^{\text{EC},2}_{i,j}$, and $b^r_{i,j}$.

Proposition 1: The optimization problem (34) is convex.

Proof: See Appendix. A

Since (34) is a convex problem, we can solve it efficiently with convex optimization tools, such as CVX [31], [32]. Similarly, we can show that (32) and (33) are convex optimization problems. Therefore, we can solve these problems using CVX. Let the optimal objective values obtained by solving (31), (32) and (33) be $\mathbb{T}_{i,j}$, $\mathbb{T}'_{m,n}$, and $\mathbb{T}_k$, respectively.

2) DUE Assignments: For a given assignment π, the optimal solution of (30) is $\max\left(\max_{\{i,j\}\in\zeta^1_\pi} \mathbb{T}_{i,j}, \max_{\{m,n\}\in\zeta^2_\pi} \mathbb{T}_{m,n}{}', \max_{k\in\rho_\pi} \mathbb{T}_k\right)$. Consequently, (30) simplifies to the following CUE-DUE assignment problem:

$$\min_{\pi\in\Pi} \max\left(\max_{\{i,j\}\in\zeta^1_\pi} \mathbb{T}_{i,j}, \max_{\{m,n\}\in\zeta^2_\pi} \mathbb{T}_{m,n}{}', \max_{k\in\rho_\pi} \mathbb{T}_k \right), \quad (35)$$

We can obtain the optimal solution of (35) by examining all possible CUE-DUE assignments $\pi \in \Pi$ and solving (31), (32) and (33) for each $\zeta^1_\pi \in \pi$, $\zeta^2_\pi \in \pi$, and ρ_π, respectively. The exhaustive search method has high computational complexity and can not be implemented in practice. We solve (35) using a low-complexity graph-theoretic matching algorithm.

We first describe some concepts of bipartite graph matching [33], [34]. A graph G that is comprised an edge set $\mathcal{E}$ and a vertex set $\mathcal{V}$ is bipartite graph if the vertex set can be partitioned into subsets $\mathcal{V}^1$ and $\mathcal{V}^2$, such that each edge $e \in \mathcal{E}$ connects a vertex in $\mathcal{V}^1$ to one in $\mathcal{V}^2$. A matching in graph G is a set of edges without common vertices. A maximum matching in graph G is a matching containing the largest possible number of edges.

Returning to (35), the following steps are used to transform the problem into a bipartite graph matching problem:

1) We represent the network as a bipartite graph such that vertices $v^1_i \in \mathcal{V}^1$ and $v^2_j \in \mathcal{V}^2$ represent CUE $i \in \{1,..,N\}$ and DUE $j \in \{1,..,M\}$, respectively.
2) For each CUE-DUE pair i, j, assign a weight to the corresponding edge (v^i_1, v^j_2) as follows: (*i.*) If $g_{i,j} > g_{i,\text{BS}}$ and $g_{\text{BS},i} > g_{\text{BS},j}$, the weight of the edge (v^1_i, v^2_j) is $\omega_{(v^i_1, v^j_2)} = \mathbb{T}_{i,j}$, where $\mathbb{T}_{i,j}$ is obtained by solving (31), (*ii.*) If $g_{i,j} < g_{i,\text{BS}}$ and $g_{\text{BS},i} > g_{\text{BS},j}$, then $\omega_{(v^i_1, v^j_2)} = \mathbb{T}'_{i,j}$, where $\mathbb{T}'_{i,j}$ is obtained by solving (32).
3) If the condition $g_{\text{BS},i} > g_{\text{BS},j}$ is not satisfied, the sum transmission rate of the BS to DUE j and BS to CUE i links is upper-bounded by $r^{\text{OMA}}_{\text{BS},j}$ and hence, the joint DUE-cloud offloading is not beneficial to DUE j. Therefore, to avoid pairing between CUE i and DUE j, the vertices v^1_i and v^2_j are not connected by an edge.
4) Next, N dummy vertices are added to $\mathcal{V}^2$ to subsume the cloud-only offloading option. Here, the edge between vertex v^1_i and the ith dummy vertex, *i.e.*, vertex v^2_{M+i}, $i \in \{1,..,N\}$, corresponds to CUE i's cloud-only offloading mode. Each edge (v^1_i, v^2_{M+i}) is assigned a weight

based on the completion time of the CUE i in cloud-only mode, *i.e.*, $\omega_{(v_i^1, v_{M+i}^2)} = \mathbb{T}_i$.

Let Φ denote the set containing all maximum matchings of this graph. Therefore, by following the above steps, the problem (35) reduces to the bottleneck matching (BM) problem of the graph, which is defined as finding the maximum matching whose largest edge weight is minimum, *i.e.:*

$$\min_{\phi \in \Phi} \max_{(v_1^i, v_2^j) \in \phi} \omega_{(v_1^i, v_2^j)}, \tag{36}$$

Since, the graph has maximum of $MN + N$ edges and $2N + M$ vertices, we can solve (36) optimally using the BM algorithm in time $\mathcal{O}(\max(N^2\sqrt{M}, M^2\sqrt{N}))$ [34]. If the output bottleneck matching contains the edge that connects vertices v_1^i and v_2^{M+i}, $i \in \{1, .., N\}$, CUE i operates in cloud-only mode. Note that the condition $g_{\text{BS},i} > g_{\text{BS},j}$ may not be satisfied for many CUE-DUE pairings in a network. Therefore, it may not be necessary to calculate $\mathbb{T}_{i,j}$ or $\mathbb{T}'_{i,j}$ for all possible CUE-DUE pairings i, j.

Let π^{p+1} be the optimal solution of (35). For each CUE-DUE pair $\{i, j\} \in \zeta^1_{\pi^{p+1}}$, let $b_{i,j}^{l,p+1}$, $b_{i,j}^{p+1}$, $b_{i,j}^{\text{EC},1,p+1}$, $b_{i,j}^{\text{EC},2,p+1}$, $b_{i,j}^{r,p+1}$, $t_{i,j}^{1,p+1}$, $t_{i,j}^{2,p+1}$, $T_{i,j}^{l,p+1}$, and $T_{i,j}^{p+1}$ be the local computation task bits, task bits offloaded to DUE j, task bits to the offloaded edge cloud at slot 1, task bits offloaded to the edge cloud at time slot 2, incentive bits, duration of the first time slot, second time slot duration, local computation delay, and computation delay at the DUE obtained by solving (31) optimally. Here, we use the notation $(\cdot)^{p+1}$ to represent the value obtained for each variable $(\cdot)$ by solving (31) at the p^{th} iteration. Let, the optimized total offloaded bits to the edge cloud and the total offload duration be denoted by $b_{i,j}^{\text{EC},p+1}$ and $t_{i,j}^{p+1}$, *i.e.*, $b_{i,j}^{\text{EC},p+1} = b_{i,j}^{\text{EC},1,p+1} + b_{i,j}^{\text{EC},2,p+1}$, $t_{i,j}^{p+1} = t_{i,j}^{1,p+1} + t_{i,j}^{2,p+1}$. For each CUE-DUE pair $\{m, n\} \in \zeta^2_{\pi^{p+1}}$, let $b_{m,n}^{l,p+1}, b_{m,n}^{p+1}, b_{m,n}^{\text{EC},p+1}, b_{m,n}^{r,p+1}, t_{m,n}^{p+1}$, $T_{m,n}^{l,p+1}$, and $T_{m,n}^{p+1}$ be the optimal values of each of the variables obtained by solving (32). Similarly, for each CUE $k \in \rho_\pi$ in cloud-only mode, $b_k^{l,p+1}$, $b_k^{\text{EC},p+1}$, t_k^{p+1}, and $T_k^{l,p+1}$ be the values of the variables obtained by solving (33) optimally. Therefore, we express the solution to (30) as $\mathcal{X}^{p+1}$, which includes π^{p+1} and the value of the variables obtained by solving (31), (32), and (33) for $\{i, j\} \in \zeta^1_{\pi^{p+1}}$, $\{m, n\} \in \zeta^2_{\pi^{p+1}}$, $k \in \rho_{\pi^{m+1}}$.

B. Cloud Resource Allocation

The terms $T_{i,j}^l$, $t_{i,j}^1 + T_{i,j}$, $T_{m,n}^l$, $t_{m,n} + T_{m,n}$, and T_k^l in the objective function of (29) as well as the constraints (29a), (29b), and (29d)–(29j) are independent of the cloud resource allocation. Consequently, (29) transforms into the following problem when the edge cloud's computing power allocated to the CUEs are variables and all other variables are fixed based on the values in $\mathcal{X}^{p+1}$:

$$\begin{aligned} \min_{V, \boldsymbol{F}} \quad & V \\ s.t. \quad & t_{i,j}^{p+1} + \frac{\beta_i b_{i,j}^{\text{EC},p+1}}{F_i} \le V \quad \{i,j\} \in \zeta_{\pi^{p+1}} \\ s.t. \quad & t_k^{p+1} + \frac{\beta_k b_k^{\text{EC},p+1}}{F_k} \le V \quad k \in \rho_{\pi^{p+1}} \\ & \sum_{l=1}^{N} F_l \le F \end{aligned} \tag{37}$$

Since (37) is a convex problem, we can solve it optimally using CVX. To reduce the number of variables in the optimization problem, the following proposition is useful.

Proposition 2: The optimal cloud resource allocation for the problem (37) can be obtained using the following steps: First, find the optimal cloud resource allocation for CUE i (which is paired with DUE j) as F_i^{p+1} by solving the following: equation:

$$F_i^{p+1} + \sum_{\substack{\{q,r\} \in \zeta_{\pi^{p+1}}, \\ \{q,r\} \neq \{i,j\}}} \frac{\beta_q b_{q,r}^{\text{EC},p+1} F_i^{p+1}}{F_i^{p+1}\left(t_{i,j}^{p+1} - t_{q,r}^{p+1}\right) + \beta_i b_{i,j}^{\text{EC},p+1}} + \sum_{k \in \rho_{\pi^{p+1}}} \frac{\beta_k b_k^{\text{EC},p+1} F_i^{p+1}}{F_i^{p+1}\left(t_{i,j}^{p+1} - t_k^{p+1}\right) + \beta_i b_{i,j}^{\text{EC},p+1}} - F = 0, \tag{38}$$

using a root-finding algorithm such as bisection search within the range $[0, F]$. Next, the optimal cloud resource allocated to each CUE q, which is paired with DUE r ($\{q, r\} \in \zeta_{\pi^{p+1}}, \{q, r\} \neq \{i, j\}$) in joint DUE-cloud offloading mode and each CUE $k \in \rho_{\pi^{p+1}}$ in cloud-only mode, are given by:

$$\begin{aligned} F_q^{p+1} &= \frac{\beta_q b_{q,r}^{\text{EC},p+1} F_i^{p+1}}{F_i^{q+1}\left(t_{i,j}^{p+1} - t_{q,r}^{p+1}\right) + \beta_i b_{i,j}^{\text{EC},p+1}} \\ F_k^{p+1} &= \frac{\beta_k b_k^{\text{EC},p+1} F_i^{p+1}}{F_i^{p+1}\left(t_{i,j}^{p+1} - t_k^{p+1}\right) + \beta_i b_{i,j}^{\text{EC},p+1}}. \end{aligned} \tag{39}$$

Proof: See Appendix. B

Let the cloud resource allocation solution in the $p+1$th iteration be $\boldsymbol{F}^{p+1} = \{F_i^{p+1}, F_k^{p+1}\}$, $i \in \zeta_{\pi^{p+1}}, k \in \rho_{\pi^{p+1}}$.

C. Iterative Algorithm

We now propose a block-coordinate descent algorithm [35] to solve (29) iteratively. At each iteration p, the problem is solved into two phases. In the first phase, problem A is solved for a fixed cloud resource allocation $\boldsymbol{F}^p$ and $\mathcal{X}^{p+1}$ is obtained. Next, the output of this phase, $\mathcal{X}^{p+1}$, is used as the input to the next step in which problem B is solved and $\boldsymbol{F}^{p+1}$ is obtained. This process is continued until the completion time objective converges. At the pth iteration, the objective value is denoted by $\mathcal{T}^N(\mathcal{X}^{p+1}, \boldsymbol{F}^{p+1})$. The detailed process is summarized in Algorithm 1.

Convergence Analysis: The convergence of Algorithm 1 is proved in the following manner. First, in Step 2, (30) is optimally solved. Hence, we have $\mathcal{T}^N(\mathcal{X}^{p+1}, \boldsymbol{F}^p) \le \mathcal{T}^N(\mathcal{X}^p, \boldsymbol{F}^p)$. Since (37) is solved optimally, we have $\mathcal{T}^N(\mathcal{X}^{p+1}, \boldsymbol{F}^{p+1}) \le \mathcal{T}^N(\mathcal{X}^{p+1}, \boldsymbol{F}^p)$. Using these steps, we show that $\mathcal{T}^N(\mathcal{X}^{p+1}, \boldsymbol{F}^{p+1}) \le \mathcal{T}^N(\mathcal{X}^p, \boldsymbol{F}^p)$ *i.e.*, objective value after each iteration is non-increasing. Furthermore, since the objective value is lower-bounded by a finite value, convergence of Algorithm 1 is guaranteed.

Computational Complexity: At each iteration in Algorithm 1, (31) or (32) is solved for a maximum of MN CUE-DUE pairs and (33) is solved for all N CUEs. Next, the bipartite graph is constructed, and BM algorithm is applied on this graph. The worst case time requirement for (31), (32) or (33) do not depend

Algorithm 1: Iterative Solution to (29).

1: Set $p = 1$ and initialize $\boldsymbol{F}^p$ according to equal computation resource allocation, *i.e.*, $F_l^p = F/N$, $l \in \{1, .., N\}$.
2: Solve (30) for given $\boldsymbol{F}^p$ following the steps described in Section V-A and represent the optimal solution as $\mathcal{X}^{p+1}$.
3: Solve (37) given $\mathcal{X}^{p+1}$ and represent the optimal solution as $\boldsymbol{F}^{p+1}$.
4: $p = p + 1$.
5: Go to step 2 and repeat until the objective function converges *i.e.*, $\mathcal{T}^N(\mathcal{X}^p, \boldsymbol{F}^p) - \mathcal{T}^N(\mathcal{X}^{p+1}, \boldsymbol{F}^{p+1}) \leq \epsilon$, $0 < \epsilon \ll 1$.

on the number of CUEs and DUEs. Hence, worst case time complexity of this step is on the order of MN. The BM algorithm has a time complexity $\mathcal{O}(\max(N^2\sqrt{M}, M^2\sqrt{N}))$. Hence, the worst case time complexity of the Algorithm 1 is determined by the time complexity of the BM algorithm. Furthermore, we have shown in Section VI that the proposed algorithm converges within a limited number of iterations. Therefore, we can conclude that the worst case computational complexity of the proposed algorithm is significantly lower compared to directly solving (29), which is non-convex for a given CUE-DUE assignment, and requires a search over $(M + N)!/M!$ number of assignments.

VI. Numerical Results

In this section, the performance of the proposed joint optimization strategy is evaluated through numerical simulations. To understand the interplay between different optimization variables, we first consider a simple network with a single CUE and and a single DUE. Then, we evaluate the performance of the proposed scheme in a multi-user setting. Furthermore, we have also compared its performance with existing related work in the case of both network settings. The fading is modeled as Rayleigh. The path-loss (in dB) is given by $37.6 \log_{10}(d[\text{km}]) + 128.1$. The simulation parameters, unless mentioned otherwise, are as follows: $P_{\text{BS}} = 45$ dBm, β_i uniformly distributed in $[500, 1500]$ cycles/bit, b_i uniformly distributed in $[200, 400]$ Kbits, $F = 20$ GHz, f_i and f_j are uniformly distributed in $[1, 3]$ GHz, $\mathsf{E}^{th} = 0.05$ J, $\gamma_c = 10^{-28}$, $N_0 = -174$ dBm/Hz, and $\epsilon = 10^{-4}$.

A. Single CUE-DUE

The distances between the CUE to BS, DUE to BS, and CUE to DUE are 80 m, 150 m, and 70 m respectively. The computation frequency of the CUE and DUE are 1 GHz and 3 GHz, respectively, $b_i = 200$ Kbits, $\beta_i = 1000$ cycles/bit, and the cloud computing frequency allocated to the CUE is 4 GHz. The uplink and downlink bandwidth are 4 MHz. Each point in the figures represents the average performance of 2000 channel realizations. Note that the proposed joint strategy is optimal for the single CUE-DUE scenario since we optimally solve (31) and (32). We compare the proposed scheme with the following benchmark scheme:

- *MEC Offloading:* The CUE operates in cloud-only mode. The completion time is minimized by deciding the computation offloading duration and the share of task bits. The optimization problem can be expressed according to (33) and can be solved using CVX. This strategy is also investigated in [11, Problem P1].

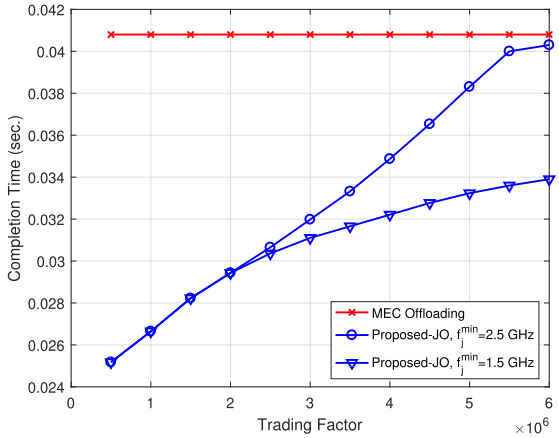


Fig. 4. Completion time vs trading factor.

In Fig. 4, we analyze the completion time for the proposed joint optimization strategy with varying trading factor when $f_j^{\min} = 1.5$ GHz and $f_j^{\min} = 2.5$ GHz. At the optimal solution, the first constraint in (31) is met with equality, and the increase in trading factor is mainly compensated by the decrease in CPU frequency at the DUE (*i.e.*, increase in computation delay at the DUE) or increase in incentive bit gain (*i.e.*, increase in the duration of transmission second slot). Therefore, the completion time increases with an increase in the trading factor using the proposed strategy. It can be observed that a significant reduction in completion time can be achieved by the proposed strategy along with providing a large incentive bit gain for the DUE when the trading factor is neither too high nor too low. For example, when $f_j^{\min} = 2.5$ GHz and the trading factor is 10^6, the completion time is reduced by 53% compared to MEC offloading scheme, and the incentive bit gain is 7.5×10^4 bits.

In Figs. 5 and 6, we examine the incentive bit gain and computation resource allocation at the DUE, respectively, to further understand the performance of the proposed strategy with varying trading factor. As the trading factor increases up to 10^6, incentive bit gain increases. Since the value of the trading factor is low in this region, a large number of computation bits can be offloaded to the DUE and computed using the allocated CPU frequency $f_j = f_j^{\max}$ at the DUE j, such that efficient management of the task computation delay is achieved at the price of a small increase in offloading delay. For higher values of the trading factor ($> 10^6$), the DUE's computation resource allocation f_j decreases with the increase in trading factor. The reason is that the value of the trading factor is high in this region, and compensating for the increase in trading factor with an increase in incentive bit gain would result in a significant increase in transmission delay (and therefore the task completion

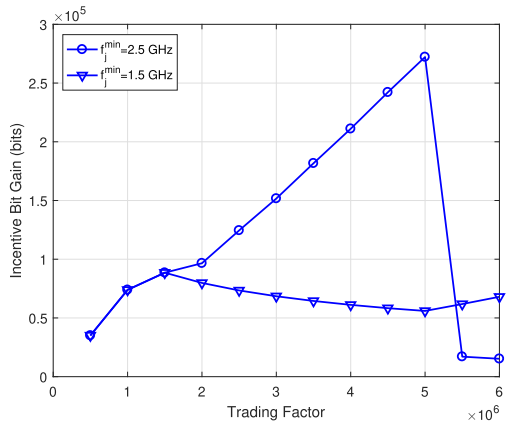


Fig. 5. Incentive bit gain vs. trading factor.

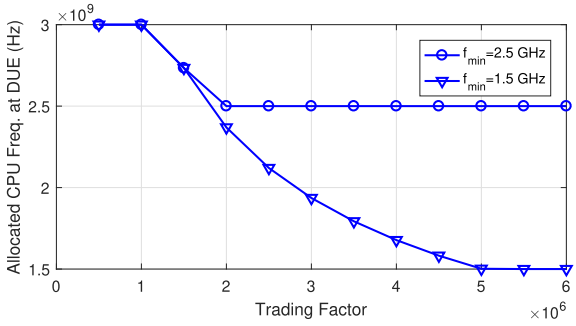


Fig. 6. Computation power allocation at the DUE vs. trading factor.

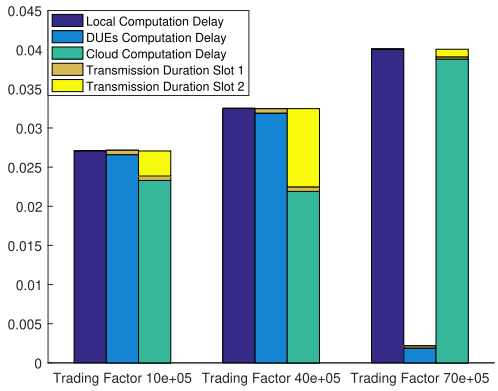


Fig. 7. Computation time and offloading delay comparison.

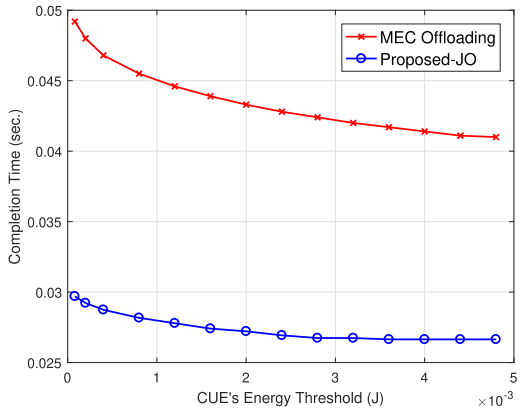


Fig. 8. Completion time vs CUE's energy threshold for task completion.

time), while compensating for the increase in trading factor with a decrease in the computation power allocation is more efficient since the term $E_{i,j}$ is related to computation resource allocation f_j as a power of 2. The completion time decreases slowly with the increase in trading factor up to $f_j = f_j^{\min}$. When $f_j^{\min} = 2.5$ GHz (or $f_j^{\min} = 1.5$ GHz) and trading factor increases beyond 2×10^6 (or 4.8×10^6), the DUE's computation resource allocation remains fixed at $f_j = f_j^{\min}$. With the increase in trading factor from 2×10^6 to 4.8×10^6 for $f_j^{\min} = 2.5$ GHz, the number of relay bits transmitted from the CUE to DUE remains high, and it increases linearly with the increase in the trading factor. Therefore, the offloading delay and task completion time of the CUE increase at the same rate. When the trading factor is greater than 4.8×10^6 and $f_j^{\min} = 2.5$ GHz, offloading large computing bits at the DUE would lead to a significant increase in incentive bits, and hence the offloading delay that can not be compensated for by the time savings of the DUE's parallel computing. Therefore, the share of task bits and the number of incentive bits sent from the CUE to the DUE approaches zero. In this case, the proposed strategy achieves the same completion time as the MEC offloading scheme.

In Fig. 7, we investigate the offloading delay and computation times for the CUE's task at each computing device for a given channel instance. The CUE to BS, DUE to BS, and CUE to DUE links have channel gains of 1.19, 0.83, and 1.22, respectively. For the MEC offloading strategy, local computation time, MEC computation time, and offloading delay are 0.0410, 0.0399, and 0.0013 sec., respectively. The computation time is well balanced across different devices in the case of both strategies. Also, the computation time is seen to dominate the offloading time, particularly for the MEC offloading strategy. We can observe that a trade-off between offloading delay and computation time at various devices is achieved by the proposed scheme. Specifically, by trading offloading delay for computation resources, the computation time of the task is reduced significantly compared to the MEC offloading strategy, and thus completion time is improved.

In Fig. 8, we evaluate the performance of the proposed scheme and MEC offloading strategy when the CUE's energy threshold for task completion increases from 8×10^{-5} to 0.0048. The task completion time decreases with the increase in the CUE's energy threshold up to 0.0024 for the proposed strategy. The task completion time of the proposed strategy remains unchanged if the CUE's energy threshold is increased further. The reason is that the energy threshold is high enough that the energy constraint in the optimization problem is not applicable. As the energy threshold increases from 8×10^{-5} to 0.0048, the reduction in CUE's task completion time compared to MEC offloading varies between 54–62%.

B. Multiple CUEs and DUEs

Here, we consider a network of size 100×100 m^2 in which an equal number of CUEs and DUEs are uniformly distributed.

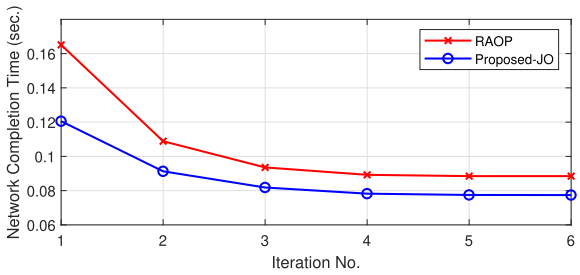


Fig. 9. Convergence analysis of Algorithm 1.

The location of the BS is $(50, 50)$. The total available uplink (downlink) bandwidth is 20 MHz, which is equally distributed among the CUEs (DUEs). The trading factor is set to 10^6. Each point in the figures represents the average performance of 200 random realizations of users' locations. In this network setting, we compare the performance of the proposed strategy with the following benchmark techniques.

1) *MEC Offloading with Cloud Resource Allocation:* Each CUE employs the cloud-only mode. In this case, the problems (33) and (37) are solved iteratively by considering $\rho_{\pi^{m+1}} = \mathcal{N}, \zeta_{\pi^{m+1}} = \emptyset$ at each iteration to obtain the final results.
2) *DUE Offloading:* Edge cloud resources are not utilized. Each CUE either computes all the task bits locally or offloads a share of its task to a DUE and sends incentive relay bits to the DUE. Selection of the DUE for each CUE, the share of task bits, relay bits, offload duration, and computation time at the CUEs and DUEs can be obtained by following the solution scheme proposed in Section V-A.
3) *Random Association Optimal Parameters (RAOP):* Here, a DUE is randomly assigned to each CUE. The values of other decision variables are obtained by iteratively solving (31), (33), and (37).
4) *NOMA Non-cooperative:* According to the transmission strategy proposed in [21], the CUEs offload their tasks over the same channel using NOMA. Specifically, we implemented the offloading solution proposed in [21, Problem (9)].

Fig. 9, shows that the proposed algorithm converges within a limited number of iterations. Note that the cloud resource is equally distributed to the CUEs in iteration 1 of Algorithm 1. We can observe that, compared to the equal cloud resource allocation, optimizing all variables jointly (6th iteration) results in a 56% reduction in the completion time.

In Fig. 10, we compare the network completion time of the proposed scheme with the benchmark schemes as the number of nodes in the network varies from 10 to 50. The number of CUEs and DUEs are equal and vary from 5 to 25. The local completion time, *i.e.,* maximum task completion time among all CUEs when each CUE's task is executed only locally at its own processor, is also shown in Fig. 10. It can be observed that the completion time increases for all the strategies with an increase in network size. The reason is that the available cloud resource per CUE decreases, and the allocated bandwidth to each node decreases, as the number of CUEs and DUEs increases. The rate

Network Size	10	20	30	40	50
Local Completion Time (Sec.)	0.43	0.48	0.50	0.51	0.52

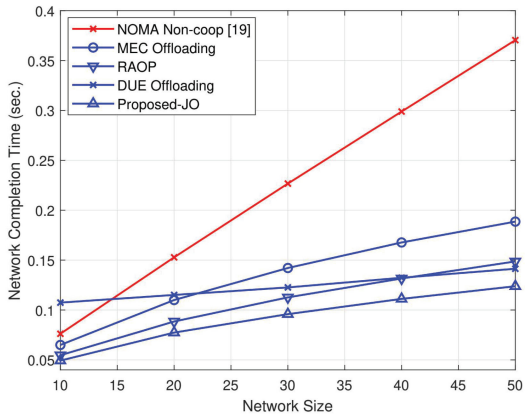


Fig. 10. Completion time vs. network size.

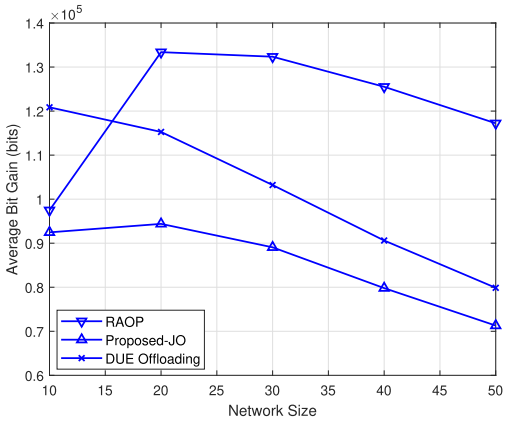


Fig. 11. Bit gain in number of bits vs. network size.

of increase in completion time for the DUE offloading scheme is slower compared to other schemes since its performance is only affected by the per node bandwidth reduction while bandwidth, as well as cloud resource per CUE, affects the performance of the other schemes. As the network size varies from 10 to 50, the reduction in the network completion time compared to the MEC offloading strategy increases from 32% to 51%. The average number of CUEs that employ joint-DUE cloud offloading mode when network size varies from 10 to 50 are 3.3, 7.5, 11.7, 16.1, and 20.6. The NOMA non-cooperative scheme has a significantly higher completion time compared to the other strategies. The reason is that the computation delay at the edge cloud is assumed to be negligible in [21], and if such an assumption does not hold, the completion time of the task at the edge cloud is not balanced across different users. Also, it can be observed that offloading the CUE's tasks to the DUEs is preferable compared to MEC offloading when the network size is greater than 20.

The average incentive bit gain and rate gain (*i.e.*, incentive bit gain per unit transmission time) performances at the DUEs with the variation of network size are demonstrated in Figs. 11 and 12, respectively. It can be observed that, on average, a large incentive bit gain (compared to the orthogonal transmission) is received

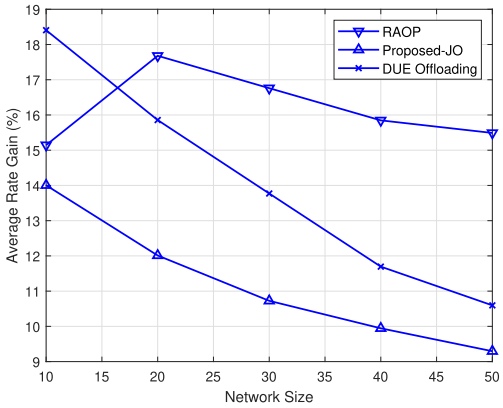


Fig. 12. Rate gain vs. network size.

TABLE I
AVERAGE ENERGY SAVING (IN PERCENTAGE) COMPARED TO MEC OFFLOADING WITH VARYING NETWORK SIZE

Scheme	10	20	30	40
RAOP	22	28	29	31
Proposed-JO	38	49	54	55

at a participating DUE using the proposed strategy, particularly when CUE-DUE assignment is random. The RAOP strategy has a higher bit gain or rate gain compared to the proposed method. The reason is that a higher bit gain (or rate gain) between a CUE-DUE pairing results in an increase in task completion time, and therefore, the assignment for which network completion time is minimized has a lower bit gain or rate gain compared to another assignment that is chosen randomly. Due to offloading a share of computation to the DUE, the number of computation bits offloaded to the edge cloud is reduced compared to the MEC offloading scheme, which results in energy saving at the edge cloud. In Table I, the average energy saving (compared to the MEC offloading) with a varying number of nodes in the network is shown. A large energy saving at the edge cloud is achieved compared to the MEC offloading strategy across different network sizes.

Next, we compare the proposed solution strategy with an empirically-obtained, globally-optimal solution. It has been shown in Section V that (29) can be solved optimally for a given cloud resource allocation. Therefore, we consider a nested loop based exhaustive search method to find the empirically-obtained, globally-optimal solution such that, in the inner loop, (29) is solved optimally for a given cloud resource allocation, and in the outer loop, a search over a very large number of cloud resource allocations is conducted. The cloud resource allocation that achieves the minimum value of $\mathcal{T}^N$ is selected as the desired solution. We consider networks of two different sizes: 1) two CUEs and two DUEs, *i.e.*, four users and 2) three CUEs and three DUEs, *i.e.*, six users. To obtain a finite set of cloud resource allocation search space, the total cloud resources are quantized into equal cloud resource chunks. The quantization parameter is denoted by Q. Then, the cloud resource allocation for four-user and six-user networks can be expressed as $\{F_1 = i \times F/Q,\ F_2 = j \times F/Q\}$ and $\{F_1 = p \times F/Q,\ F_2 = q \times F/Q,\ F_3 = r \times F/Q\}$, respectively, such that i, j in case of the four-user network or p, q, r in case of the six-user network are positive integers, and the values of these parameters are decided such that total allocated cloud resource among the CUEs is equal to F. We set $Q = 300$ and $Q = 150$ for the four-user and six-user networks, respectively. The total available cloud resources are $F = 10$ GHz. The results are averaged over 100 and 25 network realizations for the four-user and six-user networks, respectively. In Table II, we show the completion time performance of the proposed strategy and the empirically-obtained, globally-optimal solution. It can be observed that the proposed strategy performs within $1.3-2.4\%$ of the empirically-obtained, globally-optimal solution. Note that we choose the value of Q to be 300 and 150 for the four-user and six-user networks for two reasons. First, the simulation time for exhaustive search over all possible cloud resource allocations is within a reasonable time frame. Second, we have observed that the performance improvement of the empirically-obtained, globally-optimal solution when the cloud resource allocation search space is increased with larger quantization remains within 0.1% of the performance shown in Table II.

TABLE II
PERFORMANCE COMPARISON OF THE PROPOSED SCHEMES WITH EMPIRICALLY OBTAINED GLOBALLY OPTIMAL SOLUTION FOR DIFFERENT NETWORK SIZES

Netw. Size	Optimal	Proposed	RAOP	DUE-offl.
4 users	0.0398	0.0403	0.0426	0.0925
6 users	0.0526	0.0539	0.0586	0.0957

VII. CONCLUSION

We have proposed a framework for trading computation and communication resources between CUEs and DUEs that leads to a mutually-beneficial situation for the participating CUEs and DUEs, where task computation time saving at the CUE and downlink transmission rate increases at the DUE are obtained by enabling NOMA in the uplink and downlink directions. We have studied the problem of minimizing the overall completion time of the CUEs' tasks by jointly optimizing communication resource, computation resource, CUE to DUE pairing, and the share of computation and incentive bits. We have identified a suboptimum scheme that performs efficiently, and its time complexity is several orders of magnitude lower compared to the exhaustive search method. We have shown that, for a network with multiple CUEs and DUEs, the proposed strategy reduces the task completion time by 32 to 51% and provides 38 to 55% energy savings at the edge cloud compared to state-of-the-art methods. Also, a large bit gain (compared to OMA) at the DUEs can be achieved using the proposed strategy. Although these figures might change with different parameters, we expect a significant benefit in many cases of interest. Further improvements in CUEs' task completion times may be achieved by exploiting uplink NOMA such that multiple CUEs can offload their tasks within each allocated channel on top of the joint-DUE cloud offloading mode which will be investigated in future work. The proposed work can also be extended for a network with a single CUE and multiple DUEs in which the CUE's task can be

computed at multiple DUEs, and the DUEs receive information from the BS at a higher rate with the help of BS's NOMA downlink transmissions and information relaying through the CUE. The performance of such a network can be studied in the future. In this paper, queuing delay at the user devices and edge cloud is not considered, which is also the case in many prior works [7], [8], [16]–[25]. In future work, computation and communication resource optimization in the presence of queuing delay at the user devices and edge cloud can be studied.

Appendix A
Convexity Proof of (34)

The fourth constraint of (34) can be expressed using the following two constraints

$$f\left(t_{i,j}, b^r_{i,j}\right) - b^r_{i,j} - t_{i,j}c + t_{i,j} r^{\text{OMA}}_{\text{BS},j} + \mathsf{k}_j \frac{\gamma_c \beta_i^3 {b^3_{i,j}}}{{T_{i,j}}^2} \geq 0, \text{and} \tag{40a}$$

$$t^1_{i,j} + t^2_{i,j} = t_{i,j}. \tag{40b}$$

where $f(t_{i,j}, b^r_{i,j}) = t_{i,j} B_d \log\left(c_1 + c_2 2^{\frac{b^r_{i,j}}{t_{i,j} B_d}}\right)$ with $c_1 = N_0 g_{\text{BS},i} - N_0 g_{\text{BS},j}$ and $c_2 = N_0 g_{\text{BS},j}$, and $c = B_d \log(N_0 g_{\text{BS},i} + P_{\text{BS}} g_{\text{BS},i} g_{\text{BS},j})$. It is known that the perspective operation preserves convexity [36]; that is, if $g(x)$ is a convex function, then so is its perspective function $tg(x/t)$ when $t > 0$. It can be shown by second order derivative test that $B_d \log\left(c_1 + c_2 2^{\frac{b^r_{i,j}}{B_d}}\right)$ is a convex function with respect to $b^r_{i,j}$. Since $f(t_{i,j}, b^r_{i,j})$ can be obtained by applying perspective operation on $B_d \log\left(c_1 + c_2 2^{\frac{b^r_{i,j}}{B_d}}\right)$, it is a convex function. Similarly, the term $b^3_{i,j}/{T_{i,j}}^2$ can be obtained by using the perspective operation on convex function $b^3_{i,j}$, and therefore the fifth term in (40a) is convex. The second, third, and fourth terms in (40a) are linear functions of $b^r_{i,j}$, $t_{i,j}$, and $t_{i,j}$, respectively. Therefore, the fourth constraint in (34) is convex. By following proof of lemma 1 [24], we can show that sum of the first two terms of the fifth constraint in (34) is convex. The third, fourth and fifth terms can be obtained by using perspective operation on the functions $\frac{N_0}{g_{i,j}} f\left(\frac{b^r_{i,j}}{B_u}\right)$, $\frac{N_0}{g_{i,\text{BS}}} f\left(\frac{b^{\text{EC},2}_{i,j}}{B_u}\right)$, and $\gamma_c \beta_i^3 {b^l_{i,j}}^3$, respectively, and therefore they are convex functions. Hence, the fifth constraint is convex. The other constraints in (34) are linear functions. Hence, (34) is a convex optimization problem.

Appendix B
Proof of Proposition 2

It can be observed that the constraints in (37) must be satisfied at equality for the optimal solution. Then, we have

$$\begin{aligned} t^{m+1}_{i,j} + \frac{\beta_i b^{\text{EC},m+1}_{i,j}}{F^{m+1}_i} &= t^{m+1}_{p,q} + \frac{\beta_p b^{\text{EC},m+1}_{p,q}}{F^{m+1}_p} \\ &= t^{m+1}_k + \frac{\beta_k b^{\text{EC},m+1}_k}{F^{m+1}_k} \end{aligned} \tag{41}$$

for $\{i,j\}, \{p,q\}, \in \zeta_{\pi^{m+1}}, \{i,j\} \neq \{p,q\}, k \in \rho_{\pi^{m+1}}$, and,

$$\sum_{\{i,j\} \in \zeta_{\pi^{m+1}}} F_i + \sum_{k \in \rho_{\pi^{m+1}}} F_k - F = 0 \tag{42}$$

Therefore, based on (41) and after carrying out simple algebraic manipulations, we obtain the results in (39). Next, by using the results in (39) into (42), we obtain (38).

References

[1] S. Gupta, D. Rajan, and J. Camp, "NOMA enabled computation and communication resource trade-off for mobile edge computing," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–6.

[2] 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.

[3] CPU Benchmarks. [Online]. Available: https://www.cpubenchmark.net/

[4] X. Zhang, X. Hu, L. Zhong, S. Shirmohammadi, and L. Zhang, "Cooperative tile-based 360° panoramic streaming in heterogeneous networks using scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 1, pp. 217–231, Jan. 2020.

[5] J. Chakareski and S. Gupta, "Multi-Connectivity and edge computing for ultra low latency lifelike virtual reality," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2020, pp. 1–6.

[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] 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. Info. Theory*, 2017, pp. 2513–2517.

[8] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[9] 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 Comm.*, vol. 18, no. 3, pp. 1750–1763, Mar. 2019.

[10] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4188–4200, Jun. 2019.

[11] H. Xing, L. Liu, J. Xu, and A. Nallanathan, "Joint task assignment and resource allocation for D2D-enabled mobile-edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4193–4207, Jun. 2019.

[12] Y. Yang, Z. Liu, X. Yang, K. Wang, X. Hong, and X. Ge, "POMT: Paired offloading of multiple tasks in heterogeneous fog networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8658–8669, Oct. 2019.

[13] S. Gupta and A. Lozano, "Computation-bandwidth trading for mobile edge computing," in *Proc. IEEE Consum. Commun. Netw. Conf.*, 2019, pp. 1–6.

[14] M. Liwang, S. Dai, Z. Gao, Y. Tang, and H. Dai, "A truthful reverse-auction mechanism for computation offloading in cloud-enabled vehicular network," *IEEE Internet Things J.*, vol. 6, pp. 4214–4227, Jun. 2019.

[15] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.

[16] Z. Ding, P. Fan, and H. V. Poor, "Impact of non-orthogonal multiple access on the offloading of mobile edge computing," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 375–390, Jan. 2019.

[17] Z. Ding, D. W. K. Ng, R. Schober, and H. V. Poor, "Delay minimization for NOMA-MEC offloading," *IEEE Signal Process. Lett.*, vol. 25, no. 12, pp. 1875–1879, Dec. 2018.

[18] Z. Ding, J. Xu, O. A. Dobre, and H. V. Poor, "Joint power and time allocation for NOMA-MEC offloading," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6207–6211, Jun. 2019.

[19] K. Wang, Z. Ding, D. K. C. So, and G. K. Karagiannidis, "Stackelberg game of energy consumption and latency in MEC systems with NOMA," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2191–2206, Apr. 2021.

[20] Y. Wu, L. P. Qian, K. Ni, C. Zhang, and X. Shen, "Delay-minimization nonorthogonal multiple access enabled multi-user mobile edge computation offloading," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 392–407, Jun. 2019.

[21] F. Fang, Y. Xu, Z. Ding, C. Shen, M. Peng, and G. K. Karagiannidis, "Optimal resource allocation for delay minimization in NOMA-MEC networks," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7867–7881, Dec. 2020.

[22] Y. Pan, M. Chen, Z. Yang, N. Huang, and M. Shikh-Bahaei, "Energy-efficient NOMA-based mobile edge computing offloading," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 310–313, Feb. 2019.
[23] C. Li, H. Wang, and R. Song, "Intelligent offloading for NOMA-assisted MEC via dual connectivity," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2802–2813, Feb. 2021.
[24] Y. Huang, Y. Liu, and F. Chen, "NOMA-aided mobile edge computing via user cooperation," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2221–2235, Apr. 2020.
[25] S. S. Yilmaz and B. Özbek, "Multi-helper NOMA for cooperative mobile edge computing," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2021.3116421.
[26] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Comm.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.
[27] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Resource allocation for information-centric virtualized heterogeneous networks with in-Network caching and mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11339–11351, Dec. 2017.
[28] L. Yang, J. Cao, S. Tang, T. Li, and A. T. S. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, 2012, pp. 794–802.
[29] Z. Ding *et al.*, "Application of non-orthogonal multiple access in LTE and 5G networks," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 185–191, Feb. 2017.
[30] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *Proc. IEEE Veh. Technol. Conf. Spring*, 2013, pp. 1–5.
[31] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, Version 2.1.," Mar. 2014. [Online]. Available: http://cvxr.com/cvx
[32] M. Grant and S. Boyd, "The CVX users. guide, Release 2.2, CVX Res. Inc. Tech. Rep.," Jan. 2020. [Online]. Available: http://cvxr.com/cvx/doc/CVX.pdf
[33] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Philadelphia, PA, USA: SIAM, 2009.
[34] A. P. Punnen and K. P. K. Nair, "Improved complexity bound for the maximum cardinality bottleneck bipartite matching problem," *Discrete Appl. Math.*, vol. 55, no. 1, pp. 91–93, 1994.
[35] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imag. Sci.*, pp. 1758–1789, 2013.
[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

Sabyasachi Gupta received the M.Tech. degree from the National Institute of Technology, Durgapur, India, and the Ph.D. degree from the Indian Institute of Technology Delhi, New Delhi, India. He has held Research positions with the University of Pompeu Fabra, Barcelona, Spain, and with the University of Alabama, Tuscaloosa, AL, USA. He is currently a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Southern Methodist University, Dallas, TX, USA. His research interests include resource allocation for wireless networks, mobile-edge computing networks, investigating application of optimization technique, machine learning, and graph theory for wireless communication. He was the recipient of the Institute Gold Medal from NIT Durgapur in 2010.

Dinesh Rajan (Senior Member, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Madras, Chennai, India, and the M.S. and Ph.D. degrees in electrical and computer engineering from Rice University, Houston, TX, USA. He is currently the Department Chair and Cecil and Ida Green Professor with Electrical and Computer Engineering Department, Southern Methodist University (SMU), Dallas, TX, USA. In August 2002, he joined the Electrical Engineering Department, Southern Methodist University, as an Assistant Professor. His research interests include communications theory, wireless networks, information theory, and computational imaging. He was the recipient of the NSF CAREER Award for his work on applying information theory to the design of mobile wireless networks. He was also the recipient of the Golden Mustang Outstanding Faculty Award and the Senior Ford Research Fellowship from SMU.

Joseph Camp (Member, IEEE) received the B.S. (Hons.) degree in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA, and the M.S. and Ph.D. degrees in electrical and computer engineering from Rice University, Houston, TX. He is currently a Professor of electrical and computer engineering with Southern Methodist University (SMU), Dallas, TX, USA. In 2009, he joined the SMU Faculty. His research team has performed more than 200 million in-field wireless measurements around the world via Android deployment and local characterization via drones, campus buses, vehicles, and buildings. His research interests include wireless communications and networking, crowdsourcing, and drones, specifically focused on the deployment, measurement and analysis of large-scale systems, and development of embedded protocols. He was the recipient of the Ralph Budd Award for the best engineering thesis at Rice University (2010), the National Science Foundation CAREER Award (2012), the Golden Mustang Teaching Award (2014), and the Gerald J. Ford Research Fellowship (2021).