NOMA Enabled Computation and Communication Resource Trade-off for Mobile Edge Computing

Sabyasachi Gupta, Dinesh Rajan, Joseph Camp
Southern Methodist University (SMU), Dallas, TX 75275, USA.
Email: {sabyasachig, rajand, camp}@smu.edu

2021 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-7281-9505-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/WCNC49053.2021.9417425

***Abstract*—In this paper, we investigate a mobile edge computing (MEC) system in which a set of users with intensive computation tasks, and a set of users with high downlink rate requirement, can cooperate to achieve a mutually-beneficial situation where the task completion time is reduced and the downlink users receive more information from the base station (BS). Specifically, by leveraging uplink and downlink non-orthogonal multiple access (NOMA), the user with an intensive computation task can offload its task bits to the edge cloud and the downlink user. Simultaneously this user relays information to the downlink user from the BS. We consider the joint optimization of computational resource allocation at the edge cloud, communication resource allocation, assignment among the two sets of users, the share of computation, and relay bits to minimize the overall completion time of the tasks while guaranteeing downlink users' incentive requirement. A low complexity iterative algorithm is proposed to find efficient locally optimal solutions by utilizing convex optimization, a graph theory matching algorithm, and block coordinate descent technique. Numerical results show that the proposed technique leads to a significant reduction in users' task completion time and increase in the downlink users rate.**

I. Introduction

Mobile-edge computing (MEC) has emerged as a promising solution to address ever-increasing user demand for advanced applications with high computational load on mobile platform [1]. In MEC, cloud facilities are available at the edge of radio access networks, in close proximity to the mobile users, such that tasks can be computed with ultra-low latency.

The computational capacity at the edge cloud is finite. Therefore, with rapid growth in the number of computing user equipments (CUEs) that have delay-sensitive tasks to compute, offloading the task to the edge cloud may not be always beneficial. Over the past few years, the computational capability of mobile devices has increased steadily to where the performance of a mid-range mobile processor (*e.g.*, Intel Atom x5-Z83xx), is already 10% that of a edge-cloud processor (*e.g.*, Intel Xeon D-15xx) [2]. Since a large number of mobile devices in the network may not be fully utilizing their computational capabilities, offloading tasks to these peers is an enticing choice, particularly if the channel gain to these users are high. Recently, task offloading to mobile peers has been explored and joint computational resource allocation, mobile peer selection, and sharing of offloaded workload jointly optimized [3]–[5]. These studies assume that helpers are willing to compute the task for CUEs without any incentive, which may not be the case if the helping mobile peers have limited battery energy supply. For vehicular edge computing networks, computation offloading to the neighboring vehicles in which the helping vehicles are motivated by monetary incentive are considered in [6], [7]. In [8], MEC system is investigated in which tasks are offloaded to the peer devices and bandwidth incentive is provided to the peer devices.

Recently, non-orthogonal multiple access (NOMA) has been recognized as a promising approach for improving spectral efficiency of cellular networks. NOMA allows a group of users to share the same frequency/time resources for simultaneous transmissions via different power levels and successive interference cancellation (SIC) techniques. NOMA has been applied to MEC networks to reduce latency and energy consumption of MEC offloading [9], [10]. Simultaneous offloading to helping mobile peers and edge cloud using NOMA has been investigated [10]. It has been shown that the proposed strategy can reduce energy consumption compared to the non-cooperative case and orthogonal multiple access (OMA). To the best of our knowledge, incentive design for helping peers in NOMA-enabled MEC networks has not been studied before.

In a delay sensitive application in which tasks generated at the CUEs are computationally intensive, the completion time of the tasks mainly depends on the computation time at the computing devices, instead of offloading delay. Therefore, a CUE may be willing to trade its communication resources (*e.g.*, transmission energy or offloading delay) for computation resources from potential helping users. Based on this observation, in this paper, we consider a network with multiple CUEs, each with a computational intensive task and multiple downlink user equipments (DUEs) with idle processors, that are interested to receive large files from the base station (BS). We propose a novel system, in which communication and computation resource trading between a CUE-DUE pair becomes possible by enabling uplink and downlink NOMA. By deciding computation and communication resource trading partner for each CUE, task offloading time allocation, computation share at the edge cloud and DUE, incentive bits to be relayed to DUE using the CUE, and cloud resource allocation, we show that a mutually-beneficial situation can be achieved in which the completion time of the CUEs' tasks can be reduced largely, while DUEs can receive more data compared to OMA. Furthermore, energy saving at the edge cloud is also observed.

II. System Model

Assume a wireless communication system exists where a BS integrated with an edge cloud provides computing capability to

978-1-7281-9505-6/21/$31.00 ©2021 IEEE

Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on May 24,2021 at 15:54:13 UTC from IEEE Xplore. Restrictions apply.

a set $\mathcal{N} = \{1, 2, .., N\}$ of N CUEs and also communicates to a set $\mathcal{M} = \{1, 2, .., M\}$ of M DUEs in the downlink direction. We assume that resource allocation is already performed and each CUE is allocated an uplink orthogonal bandwidth and each DUE is allocated a downlink orthogonal bandwidth[1]. Quasi-static block fading is assumed for all the wireless links in the considered system. Let f_i, $i \in \mathcal{N} \bigcup \mathcal{M}$, be the computational capability (in cycles/s) at each user equipment i. Each CUE $i \in \mathcal{N}$ have a task $\phi_i = (\beta_i, b_i)$ to compute, where b_i is the number of bits to be computed, and β_i is the required number of CPU cycles to compute 1 bit of the task. Note that the methods proposed in [13] can be applied to determine b_i and β_i. Since the DUEs are equipped with idle processors, they can assist CUEs to compute their tasks when the former receives an appropriate incentive from the latter. In particular, each CUE has the following choice of modes:

- *Cloud-Only Mode*: CUE offloads a part of its task to the edge cloud. In this case, less computational power is applied to the task. However, local communication resources are fully utilized to complete the task.
- *Joint DUE-Cloud Offloading Mode*: If a CUE has a better downlink channel compared to a DUE, the sum transmission rate to these user equipments in downlink NOMA is higher compared to direct orthogonal transmission to DUE in the downlink direction. Utilizing this fact, the CUE can first receive information intended to the DUE in downlink NOMA from the BS and then forward these incentive bits to the DUE to motivate the latter to assist the former in computing its task. The CUE sends a share of its task and the incentive bits to the DUE and another share of its task to the edge cloud in the uplink direction using NOMA. Compared to the cloud-only mode, a more efficient communication resource can be utilized by enabling NOMA in the downlink and uplink, with more computational power. However in this case, less energy is available at the CUE to complete its task, and offloading delay may become high when the CUE forwards a large number of incentive bits to the DUE. The downlink and uplink transmissions by the BS and CUE, respectively, are shown in Fig. 1.

For the latter mode, we assume that each CUE is assigned to at most one DUE, and each DUE is assigned at most one CUE to reduce the system complexity. In the next two subsections, we characterize the energy and delay in the two modes and design DUE's incentive in joint DUE-cloud offloading.

A. Joint DUE-Cloud Offloading

1) CUE's NOMA Transmission: In the uplink direction, CUE uses NOMA to offload the task to the DUE and edge cloud and to forward the incentive bits to the DUE. Let, $g_{i,j}$ and $g_{i,\text{BS}}$ be the channel gain of CUE i to DUE j and CUE i to BS links, respectively. Also, let $P_{i,j}$ and $P_{i,\text{BS}}$ be the transmit power for CUE i to DUE j and CUE i to the BS

[1]Bandwidth allocation for MEC offloading is investigated previously [11], [12], and our proposed framework can be applied along with these schemes.

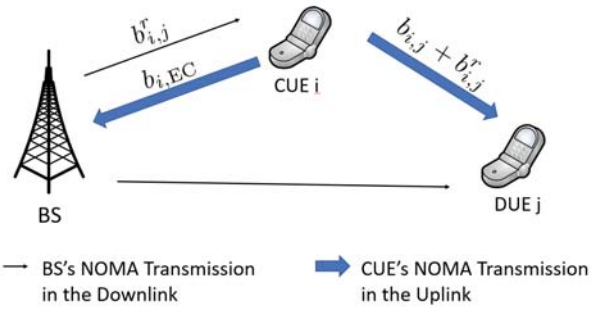

Fig. 1: Joint DUE-cloud offloading mode

links, respectively. Therefore, the transmission rate $r_{i,j}$ from CUE i to DUE j and rate $r_{i,\text{BS}}$ from CUE i to BS links are:

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

and,

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

when $g_{i,j} > g_{i,\text{BS}}$ using SIC. Here, B is the bandwidth allocated to CUE i, and N_0 is the noise power. Let $t_{i,j}$ be the uplink NOMA transmission duration. Also, let $b_{i,\text{EC}}$ be the number of task bits of ϕ_i offloaded to the edge cloud and $b_{i,j} + b^r_{i,j}$ be the number of bits CUE i sends to the DUE j in duration of $t_{i,j}$, where $b_{i,j}$ is the number of task bits of ϕ_i to be computed at the DUE j, and $b^r_{i,j}$ is the number of incentive bits that CUE receives in the downlink NOMA from BS and forwards to the DUE. Then, using (1), (2), and the relationships $b_{i,j} + b^r_{i,j} = t_{i,j} r_{i,j}$, $b_{i,\text{EC}} = t_{i,j} r_{i,\text{BS}}$, we have:

$$P_{i,j} = \frac{N_0}{g_{i,j}} f\left(\frac{b_{i,j} + b^r_{i,j}}{t_{i,j} B}\right) \tag{3}$$

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

where $f(x) = 2^x - 1$. If $g_{i,j} < g_{i,\text{BS}}$, using SIC, $r_{i,j}$ will include interference from CUE i to the BS link, while $r_{i,\text{BS}}$, will include noise only. In this case, the transmit power can be obtained in a similar manner and the optimization solution to joint DUE-cloud offloading mode can be derived following the steps in Section IV. We omit this case due to space limitation.

2) Delay and Energy for CUEs: The CUE i computes $(b_i - b_{i,j} - b_{i,\text{EC}})$ bits locally. The local computation time is:

$$T^j_i = \frac{\beta_i (b_i - b_{i,j} - b_{i,\text{EC}})}{f_i} \tag{5}$$

The computation power consumption at the CUE i is $p^j_i = \gamma_c \beta_i {f_i}^3$, where γ_c is the scaling coefficient [4]. Thus, the computation energy at CUE i is

$$E^j_i = p^j_i T^j_i = \gamma_c \beta_i (b_i - b_{i,j} - b_{i,\text{EC}}) {f_i}^2 \tag{6}$$

Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on May 24,2021 at 15:54:13 UTC from IEEE Xplore. Restrictions apply.

Let F_i be the edge cloud's processing power allocated to CUE i. The cloud has a total processing power F, *i.e.*, $\sum_{i=1}^{N} F_i \leq F$. Then, the computation delay of the shares of ϕ_i at the edge cloud and DUE j are

$$T_{\text{EC}}^{i} = \frac{\beta_i b_{i,\text{EC}}}{F_i} \tag{7}$$

and

$$T_{j}^{i} = \frac{\beta_i b_{i,j}}{f_j} \tag{8}$$

The overall completion times at DUE j and at the edge cloud are $t_{i,j} + T_j^i$ and $t_{i,j} + T_{\text{EC}}^i$, respectively. Hence, the overall completion time for task ϕ_i is:

$$\mathcal{T}_{i,j} = \max\left(T_i^j, t_{i,j} + T_j^i, t_{i,j} + T_{\text{EC}}^i\right). \tag{9}$$

We disregard the time spent in sending back the results of the computations, as the size of the output data tends to be small relative to the input data [11].

Using (3), (4), and (6), overall energy consumption to complete the task ϕ_i can be expressed as

$$\mathcal{E}_{i,j} = t_{i,j} N_0 \left(\left(\frac{1}{g_{i,\text{BS}}} - \frac{1}{g_{i,j}}\right) f\left(\frac{b_{i,\text{EC}}}{t_{i,j}B}\right) + \frac{1}{g_{i,j}} f\left(\frac{b_{i,j} + b_{i,j}^r + b_{i,\text{EC}}}{t_{i,j}B}\right)\right) + \gamma_c \beta_i (b_i - b_{i,j} - b_{i,\text{EC}}) {f_i}^2 \tag{10}$$

3) BS's NOMA Transmission: Let $g_{\text{BS},i}$ and $g_{\text{BS},j}$ be the BS to CUE i channel gain and BS to DUE j channel gain, respectively. The downlink NOMA achievable rate of BS to CUE i and BS to DUE j links can be expressed as

$$r_{\text{BS},i} = B \log\left(1 + \frac{\alpha P_{\text{BS}} g_{\text{BS},i}}{N_0}\right) \tag{11}$$

and,

$$r_{\text{BS},j} = B \log\left(1 + \frac{(1-\alpha) P_{\text{BS}} g_{\text{BS},j}}{N_0 + \alpha P_{\text{BS}} g_{\text{BS},j}}\right) \tag{12}$$

respectively, in case, $g_{\text{BS},i} > g_{\text{BS},j}$. Here, $\alpha \in (0,1)$ is the power allocation coefficient at the BS for downlink transmission. The duration of downlink NOMA transmission is $t_{i,j}$, and the BS sends a total of $b_{i,j}^r$ bits in this duration. Therefore, using (11), we have:

$$\alpha = \frac{N_0}{P_{\text{BS}} g_{\text{BS},i}} f\left(\frac{b_{i,j}^r}{t_{i,j}B}\right) \tag{13}$$

Substituting (13) into (12), we have

$$r_{\text{BS},j} = B \log\left(\frac{N_0 g_{\text{BS},i} + P_{\text{BS}} g_{\text{BS},i} g_{\text{BS},j}}{N_0 g_{\text{BS},i} + N_0 g_{\text{BS},j} f(\frac{b_{i,j}^r}{t_{i,j}B})}\right) \tag{14}$$

4) DUE's Incentive Design: The incentive of a DUE is the increase in the number of bits in joint DUE-cloud offloading mode compared to orthogonal downlink transmission within the task offloading duration subtracted by its computational energy cost and the incentive should be non-negative for DUE's participation in joint DUE-cloud offloading mode. Therefore, the constraint can be expressed as $b_{i,j}^r + t_{i,j} r_{\text{BS},j} - t_{i,j} r_{\text{BS},j}^{\text{OMA}} - \mathsf{k}_j E_j^{i\prime} \geq 0$, where $r_{\text{BS},j}^{\text{OMA}} = B \log\left(1 + \frac{P_{\text{BS}} g_{\text{BS},j}}{N_0 g_{\text{BS},j}}\right)$ is the orthogonal downlink transmission rate of DUE j, $E_j^{i\prime} = \gamma_c \beta_i b_{i,j} {f_j}^2$ is the energy consumption of the DUE j to compute $b_{i,j}$ bits, and k_j can be regarded as the minimum compensation required in terms of number of bits per unit computing energy consumption. We refer to k_j as the energy to bit cost factor. Each DUE j can decide the value of k_j based on its residual battery energy and size of the file it is interested to receive from the BS and declare the value beforehand.

B. Cloud-Only

The total completion time of CUE k's task, $k \in \{1, .., N\}$ in cloud-only mode can be expressed as:

$$\mathcal{T}_k^c = \max\left(T_k^c, t_k + T_{\text{EC}}^{k,c}\right) \tag{15}$$

where t_k, $T_{\text{EC}}^{k,c} = \beta_k b_{k,\text{EC}}^c / F_k$, and $T_k^c = \beta_k \left(b_k - b_{k,\text{EC}}^c\right)/f_k$ are, respectively, the offloading delay, computation time at the edge cloud to compute $b_{k,\text{EC}}^c$ offloaded bits, and the local computation time at CUE k. The total energy consumption is:

$$\mathcal{E}_k^c = \frac{t_k N_0}{g_{k,\text{BS}}} f\left(\frac{b_{k,\text{EC}}^c}{t_k B}\right) + \gamma_c \beta_k (b_k - b_{k,\text{EC}}^c) {f_k}^2 \tag{16}$$

III. Problem Formulation

In a network with multiple CUEs and DUEs, a set of CUEs for which joint DUE-cloud offloading mode is more beneficial compared to the cloud-only mode, may be paired with DUEs while the rest of the CUEs can operate in the cloud-only mode, to minimize task completion time of the CUEs. To formulate the network-wide assignment decision for each CUE and DUE, we define the following sets: let π be a set partition of all users, $\mathcal{N} \cup \mathcal{M}$ in which each subset has a CUE and at most one DUE, and let Π be the set of all such possible partitions. For example, with $\mathcal{N} = \{1, 2\}$ and $\mathcal{M} = \{1\}$, we have three partitions $\Pi = \Big\{\Big\{\{1,1\},\{2\}\Big\}, \Big\{\{1\},\{2,1\}\Big\}, \Big\{\{1\},\{2\}\Big\}\Big\}$. In each subset, the first and second terms are the CUE and DUE, respectively. For instance, $\big\{\{1,1\},\{2\}\big\}$ means that CUE 1 operates in joint DUE-cloud offloading mode with DUE 1, while CUE 2 operates in cloud-only offloading mode. Let ρ_π and ζ_π denote the collections of all the subsets of π with cardinalities one and two, respectively. For instance, if $\pi = \big\{\{1\},\{2,1\}\big\}$, we have $\rho_\pi = \{1\}$ and $\zeta_\pi = \{2,1\}$. We use the term CUE-DUE assignment to refer the selection of joint DUE-cloud mode and cloud-only mode for each CUE according to a member $\pi \in \Pi$.

In this paper, our aim is to minimize the maximum task completion time among all the CUEs. This maximum task completion time, $\mathcal{T}^N$, can be expressed as $\mathcal{T}^N = \max\left(\max_{\{i,j\} \in \zeta_\pi} \mathcal{T}_{i,j}, \max_{k \in \rho_\pi} \mathcal{T}_k^c\right)$. With this aim, the problem of deciding which CUEs to operate in cloud-only mode, selection of DUE for each CUE that operate in joint DUE-cloud offloading mode, the task shares to offload in two modes, the number of incentive bits transmitted to the paired

Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on May 24,2021 at 15:54:13 UTC from IEEE Xplore. Restrictions apply.

DUEs, transmission time allocation and edge cloud's resources allocation among the CUEs can be formulated as

$$\begin{aligned}\min_{\pi\in\Pi,\boldsymbol{F},\boldsymbol{b_\pi},\boldsymbol{t_\pi}} \quad & \mathcal{T}^N \\ \text{s.t.} \quad & b^r_{i,j}+t_{i,j}r_{\text{BS},j}-t_{i,j}r^{\text{OMA}}_{\text{BS},j}-\mathsf{k}_jE^{i\prime}_j\geq 0,\ \forall\{i,j\}\in\zeta_\pi \\ & \mathcal{E}_{i,j}\leq\mathcal{E}_{\text{th},i},\quad \forall\{i,j\}\in\zeta_\pi \\ & \mathcal{E}^c_k\leq\mathcal{E}_{\text{th},k},\quad \forall k\in\rho_\pi \\ & \sum_{l=1}^{N}F_l\leq F \end{aligned} \tag{17}$$

Here $\boldsymbol{b_\pi}$ is the vector of all values of $b_{i,j}, b_{i,\text{EC}}, b^r_{i,j}$, and $b^c_{k,\text{EC}}$, $\boldsymbol{t_\pi}$ is the vector of all values of $t_{i,j}$ and t_k for $\{i,j\}\in\zeta_\pi$, $k\in\rho_\pi$, and $\boldsymbol{F}$ is the vector of all values of F_l, for $l\in\mathcal{N}$. DUE's incentive requirement is captured in the first constraint. The second and third constraints ensure that the energy consumption of each CUE is bounded by an energy threshold. The edge cloud's computation resource allocated to the CUEs is restricted by F, as captured by the fourth constraints. The above optimization problem is hard to solve for two reasons: i). Given an assignment $\pi\in\Pi$, (17) is non-convex, ii). The assignment problem requires exhaustive search over a very large number of assignments. Next, we propose an efficient low-complexity sub-optimal solution.

IV. Proposed Solution

To make (17) tractable, we split it up into two optimization sub-problems: *A.* Transmission time allocation, sharing of computation task and incentive bits, and CUE-DUE assignment optimization, and *B.* Cloud resource allocation, solve them optimally at each iteration, and then iterate between these two sub-problems to converge to a final solution. Note that, "m" denotes the iteration index of the proposed algorithm.

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

At each iteration m, we solve (17) for a given cloud resource allocation $\boldsymbol{F}^m=[F^m_1,..,F^m_N]$ where F^m_i is the cloud resource allocated to CUE i, $i\in\{1,..,N\}$, at iteration m. The optimization problem can be expressed as:

$$\begin{aligned}\min_{\pi\in\Pi,\boldsymbol{b_\pi},\boldsymbol{t_\pi}} \max\left(\max_{\{i,j\}\in\zeta_\pi}\mathcal{T}_{i,j},\max_{k\in\rho_\pi}\mathcal{T}^c_k\right) \\ \text{s.t.} \quad b^r_{i,j}+t_{i,j}r_{\text{BS},j}-t_{i,j}r^{\text{OMA}}_{\text{BS},j}-\mathsf{k}_jE^{i\prime}_j\geq 0,\forall\{i,j\}\in\zeta_\pi \\ \mathcal{E}_{i,j}\leq\mathcal{E}_{\text{th},i},\quad \forall\{i,j\}\in\zeta_\pi \\ \mathcal{E}^c_k\leq\mathcal{E}_{\text{th},k},\quad \forall k\in\rho_\pi \end{aligned} \tag{18}$$

Here, the objective function is calculated based on the fixed cloud resource allocation $\boldsymbol{F}^m$. Before we present the optimal solution to (18) with all variables included, we first discuss its solution procedure for a given CUE-DUE assignment π. In this case, (18) reduces to the following independent sub-problems

$$\begin{aligned}\min_{t_{i,j},b_{i,j},b_{i,\text{EC}},b^r_{i,j}} \quad & \mathcal{T}_{i,j} \\ \text{s.t.} \quad & b^r_{i,j}+t_{i,j}r_{\text{BS},j}-t_{i,j}r^{\text{OMA}}_{\text{BS},j}-\mathsf{k}_jE^{i\prime}_j\geq 0 \\ & \mathcal{E}_{i,j}\leq\mathcal{E}_{\text{th},i} \end{aligned} \tag{19}$$

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

$$\min_{t_k,b^c_{k,\text{EC}}} \mathcal{T}^c_k,\quad \text{s.t.}\quad \mathcal{E}^c_k\leq\mathcal{E}_{\text{th},k} \tag{20}$$

for all $k\in\rho_\pi$. In the next subsection, we first present the solution to each of these independent problems. By leveraging these solutions, the optimal solution to (18) is obtained.

1) Optimal Task Offloading Time, Sharing of Computation Task and Incentive Bits: Using (5), (7)-(10), (14), the optimization problem in (19) can be expressed as

$$\begin{aligned}\min_{\substack{V,t_{i,j},b_{i,j},\\ b_{i,\text{EC}},b^r_{i,j}}} \quad & V \\ s.t.\ & \frac{\beta_i(b_i-b_{i,j}-b_{i,\text{EC}})}{f_i}\leq V \\ & t_{i,j}+\frac{\beta_ib_{i,j}}{f_j}\leq V \\ & t_{i,j}+\frac{\beta_ib_{i,\text{EC}}}{F^m_i}\leq V \\ & b^r_{i,j}+t_{i,j}B\log\left(\frac{N_0g_{\text{BS},i}+P_{\text{BS}}g_{\text{BS},i}g_{\text{BS},j}}{N_0g_{\text{BS},i}+N_0g_{\text{BS},j}f(\frac{b^r_{i,j}}{t_{i,j}B})}\right) \\ & -t_{i,j}r^{\text{OMA}}_{\text{BS},j}-\mathsf{k}_j\gamma_c\beta_ib_{i,j}f_j^2\geq 0 \\ & t_{i,j}N_0\left(\left(\frac{1}{g_{i,\text{BS}}}-\frac{1}{g_{i,j}}\right)f\left(\frac{b_{i,\text{EC}}}{t_{i,j}B}\right)\right. \\ & \left.+\frac{1}{g_{i,j}}f\left(\frac{b_{i,j}+b^r_{i,j}+b_{i,\text{EC}}}{t_{i,j}B}\right)\right) \\ & +\gamma_c\beta_i(b_i-b_{i,j}-b_{i,\text{EC}})f_i{}^2\leq\mathcal{E}_{\text{th},i} \end{aligned} \tag{21}$$

where V is a slack variable. The first three constraints of (21) are linear. It can be observed that the fourth constraint is convex since its Hessian matrix is positive semidefinite. We omit the proof due to space limits. By following the proof of lemma 1 [10], it can be shown that the fifth constraint is convex. Therefore, (21) is a convex optimization problem. Since, we have a standard convex problem, it can be solved efficiently by any convex optimization tool, such as CVX [14]. Also, it can be shown that (20), is a convex optimization problem and can be solved using CVX. Let $\mathbb{T}_{i,j}$ and $\mathbb{T}_k$ be the optimal objective value by solving (19) and (20), respectively.

2) DUE Assignments: The optimal solution of (18) can be obtained by searching over the set of all possible CUE-DUE assignments $\pi\in\Pi$ and solving (19), (20) for each $\zeta_\pi\in\pi$ and ρ_π, respectively. However, such exhaustive search is not applicable in practice due to high computational complexity. From the solutions derived in Section IV-A, (18) reduces to the simpler CUE-DUE assignment problem

$$\min_{\pi\in\Pi}\max\left(\max_{\{i,j\}\in\zeta_\pi}\mathbb{T}_{i,j},\ \max_{k\in\rho_\pi}\mathbb{T}_k\right), \tag{22}$$

which we proceed to solve optimally with low complexity by means of a graph-theoretic matching algorithm.

We summarize some concepts of bipartite graph theory matching [15]. A graph G comprising a vertex set $\mathcal{V}$ and an edge set $\mathcal{E}$ is bipartite if $\mathcal{V}$ can be partitioned into $\mathcal{V}^1$ and $\mathcal{V}^2$

Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on May 24,2021 at 15:54:13 UTC from IEEE Xplore. Restrictions apply.

(the bipartition), such that every edge in $\mathcal{E}$ connects a vertex in $\mathcal{V}^1$ to one in $\mathcal{V}^2$. A matching in G is a subset of $\mathcal{E}$ such that every vertex $v \in \mathcal{V}$ is incident to at most one edge of the matching. A maximum matching in G contains the largest possible number of edges.

We now describe the steps required to convert problem (22), into a bipartite graph matching problem:

1) The network is represented as a bipartite graph in which each CUE $i \in \{1, .., N\}$ and each DUE $j \in \{1, .., M\}$ are represented by vertices $v_i^1 \in \mathcal{V}^1$ and $v_j^2 \in \mathcal{V}^2$, respectively, and the weight of each edge (v_i^1, v_j^2) is $\omega_{(v_1^i, v_2^j)} = \mathbb{T}_{i,j}$, when $g_{\text{BS},i} > g_{\text{BS},j}$.
2) If $g_{\text{BS},i} \leq g_{\text{BS},j}$, the corresponding vertices v_i^1 and v_j^2 are not connected by an edge. The reason is, in this case, the sum transmission rate from CUE i and BS to the DUE j in joint DUE-cloud offloading mode can not be higher than BS's orthogonal transmission to DUE j, and therefore joint DUE-cloud offloading mode should not be selected between CUE i and DUE j.
3) A maximum matching for this graph corresponds to pairings between CUEs and DUEs, *i.e.*, all CUEs to operate in joint DUE-cloud offloading mode. To subsume the cloud-only offloading option, N dummy vertices are added to $\mathcal{V}^2$, with the ith dummy vertex, *i.e.*, vertex v_{M+i}^2, $i \in \{1, .., N\}$, representing the cloud-only offloading option for CUE i. The weight of each edge (v_i^1, v_{M+i}^2) is assigned as per CUE i's completion time in cloud-only mode, *i.e.*, $\omega_{(v_i^1, v_{M+i}^2)} = \mathbb{T}_i$.

By following the above steps, the CUE-DUE assignment problem (22) can be expressed as a bottleneck matching (BM) problem of the graph, which is defined by maximum matching where the largest edge weight is as small as possible, *i.e.*

$$\min_{\phi \in \Phi} \max_{(v_1^i, v_2^j) \in \phi} \omega_{(v_1^i, v_2^j)} \tag{23}$$

where Φ contains all possible maximum matchings. The bipartite graph has $2N + M$ vertices and maximum of $MN + N$ edges, and therefore, the assignment problem can be solved optimally using the BM algorithm [15] with complexity $\mathcal{O}\big(\max(N^2\sqrt{M}, M^2\sqrt{N})\big)$. In case, a vertex v_1^i, $i \in \{1, .., N\}$, is paired with its dummy vertex, *i.e.*, vertex v_2^{M+i} in the bottleneck matching of the graph, CUE i operates in cloud-only mode. Note that, for many CUE-DUE pairings in a network, a CUE may not have a better downlink channel compared to DUE, therefore calculation of $\mathbb{T}_{i,j}$ for all pairs may not be necessary.

Let π^{m+1} be the optimal solution of (22), $b_{i,j}^{m+1}$, $b_{i,\text{EC}}^{m+1}$, $b_{i,j}^{r,m+1}$ $t_{i,j}^{m+1}$, be the solution to (19) for $\{i, j\} \in \zeta_{\pi^{m+1}}$, $b_{k,\text{EC}}^{c,m+1}$, t_k^{m+1} be the solution to (20) for $k \in \rho_{\pi^{m+1}}$. Therefore, we express the solution to (18) as $\mathcal{X}^{m+1}$ which is the set of all values of π^{m+1}, $b_{i,j}^{m+1}$, $b_{i,\text{EC}}^{m+1}$, $b_{i,j}^{r,m+1}$ $t_{i,j}^{m+1}$, $b_{k,\text{EC}}^{c,m+1}$, t_k^{m+1}, $\{i, j\} \in \zeta_{\pi^{m+1}}$, $k \in \rho_{\pi^{m+1}}$.

B. Cloud Resource Allocation

In (17), the second, third, and fourth constraints as well as the terms T_i^j, $t_{i,j} + T_j^i$, and T_k^c, for all $\{i, j\} \in \zeta_{\pi^{m+1}}$, $k \in \rho_{\pi^{m+1}}$ in the objective function are independent of cloud resource allocation. Therefore, the problem (17), with variables as cloud resource allocation, while all other variables are set according to the values in $\mathcal{X}^{m+1}$, can be expressed as

$$\begin{aligned} \min_{V, \boldsymbol{F}} \quad & V \\ s.t. \quad & t_{i,j}^{m+1} + \frac{\beta_i b_{i,\text{EC}}^{m+1}}{F_i} \leq V \quad \{i, j\} \in \zeta_{\pi^{m+1}} \\ s.t. \quad & t_k^{m+1} + \frac{\beta_k b_{k,\text{EC}}^{c,m+1}}{F_k} \leq V \quad k \in \rho_{\pi^{m+1}} \\ & \sum_{l=1}^{N} F_l \leq F \end{aligned} \tag{24}$$

The above problem is convex and can be solved optimally using CVX. The optimal cloud resource allocation solution is denoted by $\boldsymbol{F}^{m+1}$.

C. Iterative Algorithm

We now propose an iterative algorithm to solve the optimization problem (17) by using the block-coordinate descent method [16]. Here, we split up our problem into two phases i) Joint time allocation, sharing of computation task and incentive bits, CUE-DUE assignment optimization, ii) Cloud resource allocation. In each iteration m, in the first phase, we solve the problem of joint time allocation, sharing of computation task and incentive bits, CUE-DUE assignment optimization for fixed cloud resource allocation and obtain $\mathcal{X}^{m+1}$. Then, the output of this phase, $\mathcal{X}^{m+1}$, is used as an input for the next step in which we solve the cloud resource allocation problem to obtain $\boldsymbol{F}^{m+1}$. The objective value of (17) obtained at the mth iteration is denoted by $\mathcal{T}^N(\mathcal{X}^{m+1}, \boldsymbol{F}^{m+1})$. The steps are described in Algorithm 1.

To solve (17) using Algorithm 1, at each iteration, (19) needs to be solved for a maximum of MN different CUE-DUE pairs to obtain $\mathbb{T}_{i,j}$, for $i \in \{1, .., N\}$, $j \in \{1, .., M\}$ and (20) needs to be solved for N CUEs to obtain all $\mathbb{T}_i$s such that the bipartite graph can be constructed. The time complexity of each of these optimization problems is independent of parameters M or N, and therefore overall complexity of this step is on the order of MN. Next, the BM algorithm runs in time $\mathcal{O}\big(\max(N^2\sqrt{M}, M^2\sqrt{N})\big)$. Therefore, the time complexity of Algorithm 1 at each iteration is decided by the BM algorithm's time complexity. Also, it has been shown in Section V that the algorithm converges within a small number of iterations. Therefore, compared to directly solving (17) which is firstly non-convex and has a large number of variables for a given assignment π (on the order of M, N) and requires an exhaustive search over $(M + N)!/M!$ number of assignments, computation complexity of our proposed algorithm is several orders of magnitude lower.

D. Convergence Analysis

The convergence of Algorithm 1 is proved as follows. First, in Step 2, we optimally solve (18), and therefore, we have $\mathcal{T}^N(\mathcal{X}^m, \boldsymbol{F}^m) \geq \mathcal{T}^N(\mathcal{X}^{m+1}, \boldsymbol{F}^m)$.

Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on May 24,2021 at 15:54:13 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Iterative Algorithm to Solve (17)

1: Initialize $\boldsymbol{F}^m$, $m = 1$, according to equal allocation, *i.e.*, $F_l^m = F/N$, $l \in \{1, .., N\}$.
2: Find the optimal solution of the problem (18) for given $\boldsymbol{F}^m$ by following the procedure described in Section IV-A and denote the optimal solution as $\mathcal{X}^{m+1}$.
3: Solving optimization problem (24) given variables $\mathcal{X}^{m+1}$ and denote the solution as $\boldsymbol{F}^{m+1}$.
4: Update $m = m + 1$.
5: Go to Step 2 and repeat until the convergence is obtained, *i.e.*, $\mathcal{T}^N(\mathcal{X}^m, \boldsymbol{F}^m) - \mathcal{T}^N(\mathcal{X}^{m+1}, \boldsymbol{F}^{m+1}) \leq \epsilon$, $0 < \epsilon \ll 1$.

Next, since optimal solution of (24) is obtained, we have $\mathcal{T}^N(\mathcal{X}^{m+1}, \boldsymbol{F}^m) \geq \mathcal{T}^N(\mathcal{X}^{m+1}, \boldsymbol{F}^{m+1})$. Therefore, we can conclude that $\mathcal{T}^N(\mathcal{X}^m, \boldsymbol{F}^m) \geq \mathcal{T}^N(\mathcal{X}^{m+1}, \boldsymbol{F}^{m+1})$. It indicates that the objective value of Algorithm 1 after each iteration is non-increasing. In addition, we see that the objective value is lower bounded by a finite value. Hence, the proposed algorithm is guaranteed to converge.

V. Numerical Results

In this section, numerical results are provided to evaluate the performance of the proposed strategy, as compared to the following two benchmark schemes.

1) *Cloud Offloading*: Each CUE operates in cloud-only mode. The optimization for this strategy can be obtained by iteratively solving when (20) and (24), considering $\rho_{\pi^{m+1}} = \mathcal{N}$, $\zeta_{\pi^{m+1}} = \emptyset$ at each iteration m.
2) *DUE-Cloud Random*: In this case, each CUE is assigned a DUE randomly, and then other variables are optimized by iteratively solving (19), (20), and (24) for the given random assignment.

We first investigate performance of the proposed strategy and cloud offloading scheme with varying energy to bit cost factor for a single CUE-DUE scenario. Then, we will demonstrate performance of these strategies in a network with multiple CUEs and DUEs. The system parameters are $P_{\text{BS}} = 45$ dBm, channel model Rayleigh fading, pathloss coefficient 3, $N_0 = -102$ dBm, f_i, f_j uniform distributed in $[1, 3]$ GHz, b_i uniformly distributed in $[200, 400]$ Kbits, $\beta_i = 1000$ cycles/bit, and $\gamma_c = 10^{-28}$. The energy threshold for each CUE is set according to its energy requirement to compute the task locally. The total 20 MHz uplink bandwidth is equally allocated among the CUEs. The cloud power allocated to the CUE for the single CUE-DUE case is 4 GHz, and for multiple CUEs and DUEs case, total cloud power is $F = 20$ GHz.

A. Single CUE-DUE

In this scenario, the distance between the CUE to BS, CUE to DUE, BS to DUE are 80 m, 70 m, and 150 m, respectively. The results are averaged over 2000 channel realizations.

In Fig. 2, we analyze performance in terms of completion time for the proposed strategy and cloud offloading scheme with varying energy to bit cost factor. In Fig. 3, we investigate the offloaded share of computation task to the DUE, and the

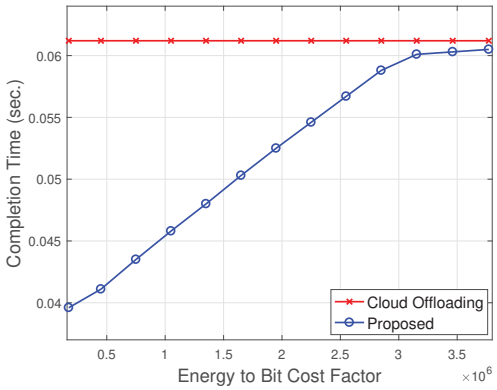

Fig. 2: Completion Time vs energy to bit cost factor

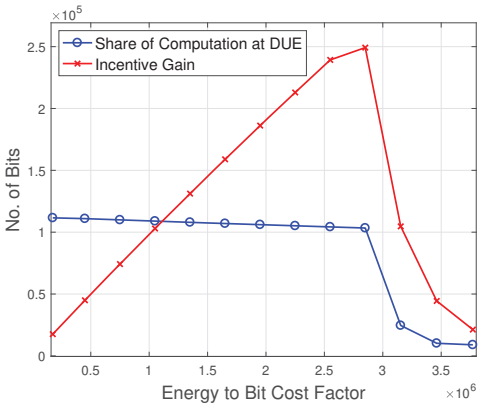

Fig. 3: Share of computation and incentive bit gain vs. energy to bit cost factor using the proposed strategy

excess bit gain received using the proposed strategy (compared to the downlink orthogonal transmission to the DUE) with varying energy to bit cost factor. As the energy to bit cost factor increases upto 2.8×10^6, the share of computational task offloaded to the DUE slowly decreases. In this case, the rate of increase in number of incentive bits relayed through the CUE is high to compensate for the high computational energy consumption with increase in energy to bit cost factor. Since the number of bits transmitted from the CUE to the DUE increases rapidly, the offloading delay and therefore, CUE's task completion time decreases in the same rate. When the energy to bit cost factor is greater than 2.8×10^6, relaying large number of incentive bits to offload large number of computing bits would result in a high offloading delay that may not be compensated by time saving with parallel computing at the DUE. Thus, share of computation at the DUE and incentive bit gain approaches zero. The completion time in this case is same as the cloud offloading scheme. We observe that the proposed strategy can reduce the completion time of the task significantly, while a large number of bit gain can be achieved for the DUE when the energy to bit cost factor is within the range 10^6 to 2×10^6.

B. Multiple CUEs and DUEs

For the evaluations that follow, 5 CUEs and DUEs are uniformly distributed in a square region of 100×100, and the BS with an edge cloud is located at $(100, 100)$. The energy to

Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on May 24,2021 at 15:54:13 UTC from IEEE Xplore. Restrictions apply.

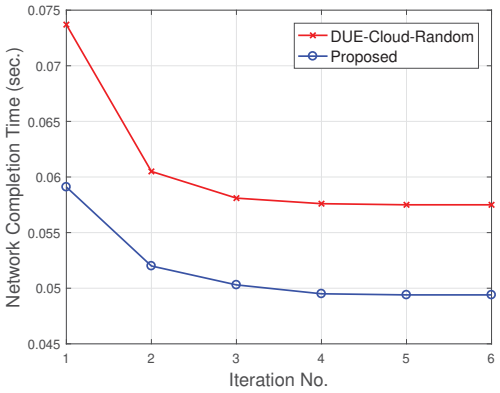

Fig. 4: Convergence analysis of the iterative algorithm

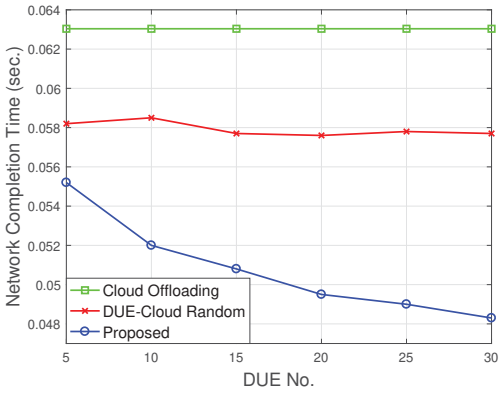

Fig. 5: Completion time vs. number of DUEs

bit cost factor is set to 10^6. The results are averaged over 200 network realizations.

In Fig. 4, we show the convergence of the proposed algorithm. The proposed algorithm can converge within a small number of iterations. Also, it can be observed that compared to the equal cloud resource allocation at iteration 1, jointly optimizing cloud resource along with all other variables in six iterations results in 20% decrease in completion time.

In Fig. 5, we compare the performance of the proposed offloading strategy with two benchmark schemes in terms of network completion time when number of DUEs in the network varies from 5 to 30. As the number of DUEs increases, more DUEs that have high CUE-DUE channel gain, and high computation power, may become available and therefore, the completion time decreases for the proposed strategy and DUE-cloud random. For a network with 30 DUEs, proposed strategy can reduce the network completion time by 30% compared to cloud offloading. Offloading computation to the DUE results in reduction in the number of bits to be computed at the edge cloud compared to the cloud offloading scheme, which leads to energy saving at the edge cloud. In Table I, we show the average energy saving with varying number of DUEs.

VI. Conclusion

In this paper, we have proposed a computation and communication resource trading strategy between CUEs and DUEs that leads to a mutually-beneficial situation for CUE's task computation and DUE's downlink transmission rate by enabling NOMA in the uplink and downlink. We have studied joint optimization of computation and communication resource allocation, assignment among the CUEs and DUEs, share of computation and incentive bits with the aim of minimizing overall completion time of the tasks. Although a complete optimization is exceedingly complex, we have identified sub-optimum approach that perform efficiently, while achieving a significant reduction in the solution complexity. We have shown, that the proposed strategy reduces the network completion time by 30%, while providing a large bit gain (compared to OMA) to the DUEs during the task offloading duration.

TABLE I: Average energy saving (in Joule) compared to cloud offloading scheme with varying number of DUEs.

Scheme	10	15	20	25
DUE-Cloud Random	0.13 J	0.16 J	0.19 J	0.18 J
Proposed	0.46 J	0.52 J	0.57 J	0.59 J

References

[1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing-A key technology towards 5G," *ETSI White Paper*, 2015.
[2] [Online]. Available: https://www.cpubenchmark.net/
[3] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE TWC*, vol. 18, no. 3, pp. 1750–1763, 2019.
[4] 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, 2019.
[5] 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, 2019.
[6] 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, 2019.
[7] 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, 2019.
[8] S. Gupta and A. Lozano, "Computation-bandwidth trading for mobile edge computing," in *Proc. IEEE CCNC*, 2019, pp. 1–6.
[9] 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, 2019.
[10] Y. Huang, Y. Liu, and F. Chen, "NOMA-aided mobile edge computing via user cooperation," *IEEE TCOM*, vol. 68, no. 4, pp. 2221–2235, 2020.
[11] 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 TWC*, vol. 16, no. 8, pp. 4924–4938, 2017.
[12] 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. 11 339–11 351, 2017.
[13] 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. Int. Conf. CLOUD*, Jun. 2012, pp. 794–802.
[14] M. Grant and S. Boyd. (2014) CVX: Matlab Software for Disciplined Convex Programming. [Online]. Available: http://cvxr.com/cvx
[15] A. P. Punnen and K. P. K. Nair, "Improved complexity bound for the maximum cardinality bottleneck bipartite matching problem," *Discrete Applied Mathematics*, vol. 55, no. 1, pp. 91 – 93, 1994.
[16] 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.*, p. 1758–1789, 2013.

Authorized licensed use limited to: SOUTHERN METHODIST UNIV. Downloaded on May 24,2021 at 15:54:13 UTC from IEEE Xplore. Restrictions apply.