Task Partitioning and User Association for Latency Minimization in Mobile Edge Computing Networks

Mingjie Feng, Marwan Krunz, and Wenhan Zhang
Dept. Electrical & Computer Engineering, The University of Arizona, Tucson, AZ 85719, USA.
Email: {mingjiefeng, krunz, wenhanzhang}@email.arizona.edu

***Abstract*—Mobile edge computing (MEC) is a promising solution to support emerging delay-sensitive mobile applications. With MEC servers deployed at the network edge, the computational tasks generated by these applications can be offloaded to edge nodes (ENs) and quickly executed there. Meanwhile, with the projected large number of IoT devices, the communication and computational resources allocated to each user can be quite limited, providing low-latency MEC services becomes challenging. In this paper, we investigate the problem of task partitioning and user association in an MEC system, aiming to minimize the average latency of all users. We assume that each task can be partitioned into multiple independent subtasks that can be executed on local devices (e.g., vehicles), MEC servers, and/or cloud servers; each user can be associated with one of the nearby ENs. We formulate a mixed-integer programming problem to determine the task partitioning ratios and user association. Such a problem is solved by decomposing it into two subproblems. The lower-level subproblem relates to task partitioning under a given user association, which can be solved optimally. The higher-level subproblem is user association, where we propose a dual decomposition-based approach to solve it. Simulation results show that, compared to benchmark schemes, the proposed schemes reduce the average latency by approximately 50%.**

***Index Terms*—Mobile edge computing; delay-sensitive IoT applications; task partitioning; user association.**

I. Introduction

The emergence of mobile Internet of Things (IoT) applications (e.g., autonomous driving, augmented/virtual reality) has triggered a growing demand for executing computationally intensive tasks with stringent delay requirements [1]. Given the limited processing capability of mobile devices, it is challenging for users to timely execute these tasks. Mobile edge computing (MEC) is a promising solution to this challenge. With MEC servers deployed at the network edge, e.g., near base stations (BSs), users can offload their computational tasks to nearby edge servers for fast execution. Benefiting from the proximity to end-users, low latency can be achieved.

At the same time, tens of billions of mobile devices will soon be connected to the Internet in the near future [2], many of which are to be supported by future MEC systems. These devices will compete for limited computing and communication resources, increase the workload of edge servers, and making it less likely that the MEC systems will deliver the expected low-latency services to all connected users [10]. To address this challenge, the optimization of task offloading decision and resource allocation among users served by an MEC server have been explored in the literature (e.g., [10], [12]). The performance of MEC systems can also be enhanced via collaboration among multiple MEC servers [9], [15], which enables computational tasks to be transferred between these servers for improved load balancing.

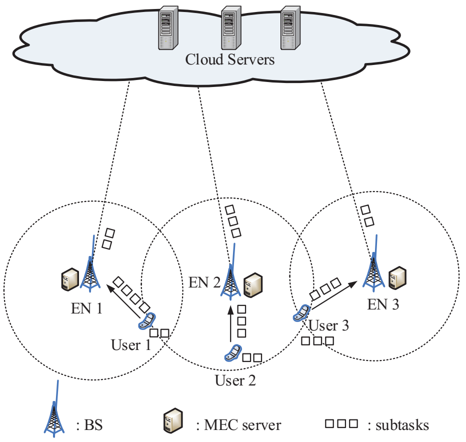


Fig. 1. System model of a multi-user MEC system with one cloud server and multiple MEC servers.

Another approach for latency reduction in MEC is task partitioning. Most existing works on task offloading assume that the computation of a task begins after the whole task has been offloaded to the MEC or cloud server. In contrast, if a task can be partitioned into multiple subtasks and assigned to the local device, the MEC server, and/or the cloud server for execution, the workload at each of these entities can be reduced. Besides, the offloading and computing processes can be performed *concurrently*, resulting in lower latency. Obviously, the task partitioning ratios need to be optimized based on various system parameters, e.g., computational capabilities of different devices/servers, channel quality between the user and edge node (EN)[1], traffic load, etc.

Task partitioning has been recently considered, based on the model of a single EN [13], [14] or a single user [15]. In an MEC system with multiple ENs serving multiple users (see Fig. 1), user association is a key design factor, as it determines the traffic load at each EN and the latency associated with offloading a task to different ENs. Thus, user association directly impacts the task partitioning strategy, necessitating a joint optimization of task partitioning and user association.

[1] Here, an EN refers to a combination of a BS/AP and an MEC server.

In this paper, we investigate the problem of optimizing task partitioning and user association to minimize the average latency of users in a cellular-network-based MEC system. We develop efficient schemes to obtain near-optimal solutions to the problem. The contributions are summarized as follows.

- We formulate the problem of joint optimization of task partitioning and user association in MEC systems. We consider the case where the tasks can be decomposed into multiple independent subtasks[2]. A mixed-integer linear programming (MILP) problem is formulated with the objective of minimizing the average latency of all users.
- We solve the formulated problem by decomposing it into two subproblems. The lower-level subproblem targets optimizing the task partitioning ratio under a given user association, which can be optimally solved. The higher-level subproblem is user association, for which we develop a dual decomposition-based scheme to obtain a near-optimal solution.
- To demonstrate the near-optimality of our solutions, we derive a lower bound on the average latency.
- We evaluate the performance of the proposed schemes via simulations. The results show that, compared to benchmark schemes, the proposed schemes reduce the average latency by around 50%.

In the remainder of this paper, we first review related literature in Section II. Then, we introduce the system model in Section III, followed by the problem formulation given in Section IV. Algorithmic solutions are presented in Section V. We present our simulation results and discussion in Section VI. Finally, the paper is concluded in Section VII.

II. Related Work

Task assignment in MEC systems was investigated in prior works (e.g., [3]–[7]). The majority of existing works are based on *binary* task assignment, where a task can either be offloaded to an MEC server or executed locally. While most existing works consider models based on homogeneous tasks, task assignment for heterogeneous tasks was recently proposed [7]. In contrast to these works, we extend the notion of task assignment to fully exploit the computational capability of local devices, MEC servers, and cloud servers by allowing individual tasks to be partitioned.

Task partitioning has been considered in some recent works under different partitioning patterns and design objectives [10]–[15], [17]. The partitioning between the local device and the cloud server was considered in [10], while task partitioning between the local device and the MEC server was considered in [11]–[14]. In [11], joint optimization of the task partitioning ratio, device transmit power, and device computational speed was performed to minimize the device's energy consumption and task execution latency. In [12], the task partitioning ratio and communication resources were optimized to minimize the total energy consumption. In [13]–[15], the optimal partitioning ratio and resource allocation were derived with the objective of minimizing the overall offloading latency. Our paper differs from the above works in that we consider task partitioning among local device, EN, and cloud server to fully utilize task partitioning for latency reduction. Moreover, compared to these works that targeted a single EN or a single user, we consider a multi-EN-multi-user setting, which necessitates optimizing user association.

To harness the benefits of utilizing multiple ENs for task offloading, cooperation among ENs was considered [8], [9], [15]. Specifically, a user can offload its tasks to multiple ENs [8], [15], or the ENs can send their workloads to each other [9]. The optimization of user association in multi-cell based MEC systems was recently investigated. In [16], [17], joint optimization of user association and resource allocation was carried out to minimize the total energy consumption. In contrast to these works, we aim to minimize the *average latency* in delay-sensitive MEC applications.

III. System Model

A. Problem Setup

We consider a multi-user MEC system consists of one cloud server and multiple MEC servers that are placed next to or integrated into the BSs of a wireless cellular network. The combination of a BS and an MEC server is regarded as an edge node (EN), which is connected to the cloud server via backhaul connections. There are J ENs indexed by $j \in \{1,\dots,J\} \triangleq \mathcal{J}$, which collectively serve K mobile users equipments (UE), indexed by $k \in \{1,\dots,K\} \triangleq \mathcal{K}$. The user associations are defined by the following binary variables:

$$x_{k,j} \triangleq \begin{cases} 1, & \text{if UE } k \text{ is associated with EN } j \\ 0, & \text{otherwise,} \end{cases} \qquad k \in \mathcal{K},\ j \in \mathcal{J}. \quad (1)$$

We consider each UE can be associated with at most one EN. For UEs associated with EN j, their tasks can be executed at EN j and/or forwarded by EN j to the cloud server for execution. We assume that each UE generates one task at a time. Each task can be partitioned into multiple independent subtasks, each with its own data. An example of such a task is object recognition from videos taken by cameras. Each video clip can be partitioned into multiple segments and processed at the UE, EN, and cloud server, respectively. Suppose that $x_{k,j}=1$, the ratios of subtasks assigned to UE k, EN j, and the cloud server are denoted by α_k, $\beta_{k,j}$, and $\gamma_{k,j}$, respectively. Specifically, they are the fractions of input data (e.g., file sizes of video segments) of the task generated by UE k.[3]

B. Computational Model

The task generated by any UE k is parameterized by the size of input data s_k (in bits) and the computational complexity z_k,

[2] The case of dependent subtasks will be investigated in future work.

[3] Because a task cannot be partitioned into arbitrarily small subtasks, $(\alpha_k,\beta_{k,j},\gamma_{k,j})$ can only take a finite number of values. For example, if a task can be partitioned into 10 comparable subtasks, the partitioning ratios can only be values in the set of $\{0, 0.1, \dots, 0.9, 1\}$. In this paper, we first obtain the optimal $(\alpha_k,\beta_{k,j},\gamma_{k,j})$ in the continuous domain $[0,1]$ and then round them to the closest feasible values.

defined by the number of CPU cycles required to execute one bit of the task. Then, the number of CPU cycles required to complete the whole task is $s_k z_k$.

1) Local UE Computing Time: Let $c_k^{(\mathrm{L})}$ be the computational capability of UE k, measured in CPU cycles per second. Given α_k, the number of CPU cycles required to complete the subtasks assigned to UE k is $\alpha_k s_k z_k$. Then, the execution time in seconds at UE k is given by:

$$t_{\mathrm{comp},k}^{(\mathrm{L})} = \frac{\alpha_k s_k z_k}{c_k^{(\mathrm{L})}}. \tag{2}$$

2) MEC Server Computing Time: Let $c_j^{(\mathrm{E})}$ be the computational capability of EN j. We assume that this capability is equally allocated to all UEs associated with EN j during each time slot. For notational simplicity, we denote the traffic load of EN j by $Q_j \triangleq \sum_{k=1}^{K} x_{k,j}$. Given $\beta_{k,j}$, the execution time for the subtasks of UE k at EN j is given by:[4]

$$t_{\mathrm{comp},k,j}^{(\mathrm{E})} = \frac{\beta_{k,j} s_k z_k}{c_{k,j}^{(\mathrm{E})}} = \frac{\beta_{k,j} s_k z_k Q_j}{c_j^{(\mathrm{E})}}, k \in \mathcal{K},\ j \in \mathcal{J}. \tag{3}$$

3) Cloud Server Computing Time: We assume that the cloud server provides a fixed computational capability to UE k, given by $c_k^{(\mathrm{C})}$, which is based on the plan of service purchased by UE k. Suppose UE k is associated with EN j, the execution time at the cloud server is given by:

$$t_{\mathrm{comp},k,j}^{(\mathrm{C})} = \frac{\gamma_{k,j} s_k z_k}{c_k^{(\mathrm{C})}}, k \in \mathcal{K},\ j \in \mathcal{J}. \tag{4}$$

C. Communication Model

The cellular network considered in this paper adopts an orthogonal time-frequency resource allocation, e.g., OFDMA, as used in LTE and 5G systems. The communication resource is equally allocated among all UEs associated with an EN to achieve logarithmic rate maximization [18]. Each EN can measure the uplink signal-to-interference-plus-noise ratio (SINR) of UEs associated with it [12]–[14]. Then, the data rate of UE k when associated with EN j is given by:

$$R_{k,j} = \frac{W \log\left(1+\theta_{k,j}\right)}{Q_j} \tag{5}$$

where W is the bandwidth of the access channel for each EN and $\theta_{k,j}$ is the SINR for the uplink from UE k to EN j. We assume that UE k can be connected to EN j only when $\theta_{k,j}$ is no less than a given threshold θ_{th}. Let π_k be the set of ENs that can be employed by UE k for task offloading, $\pi_k = \{j \,|\theta_{k,j} \geq \theta_{\mathrm{th}}\}$. Then, we have:

$$x_{k,j} = 0, \forall j \notin \pi_k. \tag{6}$$

We assume that a UE only needs to upload the input data of its subtasks to the associated EN. The subtasks to be offloaded include the ones to be executed by the EN and the cloud server. Then, the offloading time from UE k to EN j is given by:

$$t_{\mathrm{off},k,j}^{(\mathrm{E})} = \frac{(\beta_{k,j}+\gamma_{k,j}) s_k}{R_{k,j}} = \frac{(\beta_{k,j}+\gamma_{k,j}) s_k Q_j}{W \log\left(1+\theta_{k,j}\right)}. \tag{7}$$

We consider a wired backhaul link of rate M_j between EN j and the cloud sever, and the link capacity is equally divided to transmit the input data of all UEs served by EN j. Then, the backhaul transmission time is $t_{\mathrm{off},k,j}^{(\mathrm{B})} = \frac{\gamma_{k,j} s_k Q_j}{M_j}$. Thus, the total time required for offloading the subtasks of UE k to the cloud server via EN j is given by:

$$t_{\mathrm{off},k,j}^{(\mathrm{C})} = t_{\mathrm{off},k,j}^{(\mathrm{E})} + t_{\mathrm{off},k,j}^{(\mathrm{B})}. \tag{8}$$

Due to the small size of output data, the latency for sending the outcome of a task to a UE is neglected [4], [12].

D. Task Completion Latency

Let $t_k^{(\mathrm{L})}$, $t_k^{(\mathrm{E})}$, and $t_k^{(\mathrm{C})}$ be the total elapsed time until the subtasks of UE k are completed at the local device, EN j, and cloud server, respectively, they are calculated by:

$$\begin{aligned} t_k^{(\mathrm{L})} &= t_{\mathrm{comp},k}^{(\mathrm{L})}, \\ t_{k,j}^{(\mathrm{E})} &= t_{\mathrm{off},k,j}^{(\mathrm{E})} + t_{\mathrm{comp},k,j}^{(\mathrm{E})}, \\ t_{k,j}^{(\mathrm{C})} &= t_{\mathrm{off},k,j}^{(\mathrm{C})} + t_{\mathrm{comp},k,j}^{(\mathrm{C})}. \end{aligned} \tag{9}$$

As the subtasks are independent of each other, they can be concurrently executed. Thus, the latency of for completing the whole task is the latency of *latest* completed part, given by:

$$T_k = \max\left\{ t_k^{(\mathrm{L})}, \sum_{j=1}^{J} x_{k,j} t_{k,j}^{(\mathrm{E})}, \sum_{j=1}^{J} x_{k,j} t_{k,j}^{(\mathrm{C})} \right\}. \tag{10}$$

IV. Problem Formulation

In this paper, we aim to minimize the average latency of UEs. Let $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, and $\mathbf{x}$ denote the vector $[\alpha_k]_{k\in\mathcal{K}}$, the matrix $[\beta_{k,j}]_{k\in\mathcal{K},j\in\mathcal{J}}$, the matrix $[\gamma_{k,j}]_{k\in\mathcal{K},j\in\mathcal{J}}$, and the matrix $[x_{k,j}]_{k\in\mathcal{K},j\in\mathcal{J}}$, respectively. The problem formulated as:

$$\begin{aligned} \mathbf{P1}:\ & \min_{\{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\mathbf{x}\}} \textstyle\sum_{k=1}^{K} T_k & (11)\\ \text{s.t.:}\ & \alpha_k + \textstyle\sum_{j=1}^{J}\beta_{k,j} + \sum_{j=1}^{J}\gamma_{k,j} = 1,\ k \in \mathcal{K}, & (12)\\ & \textstyle\sum_{j=1}^{J} x_{k,j} \leq 1,\ k \in \mathcal{K}, & (13)\\ & \textstyle\sum_{k=1}^{K} x_{k,j} \leq S_j,\ j \in \mathcal{J}, & (14)\\ & \beta_{k,j}, \gamma_{k,j} \leq x_{k,j},\ k \in \mathcal{K},\ j \in \mathcal{J}, & (15)\\ & 0 \leq \alpha_k, \beta_{k,j}, \gamma_{k,j} \leq 1,\ k \in \mathcal{K},\ j \in \mathcal{J}, & (16)\\ & x_{k,j} \in \{0,1\},\ k \in \mathcal{K},\ j \in \mathcal{J}, & (17)\\ & x_{k,j} = 0,\ k \in \mathcal{K},\ \forall j \notin \pi_k. & (18) \end{aligned}$$

In problem **P1**, constraints in (12) come directly from the definitions of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$; constraints in (13) indicate that each UE can be associated with at most one EN; constraints in (14) enforce the upper bound on the number of UEs that can be served by EN j, given by S_j (e.g., S_j is the number of channels); constraints in (15) are due to the fact that a UE

[4] We assume that the computation resource allocated to each task remains the same within a time slot. When some tasks are completed earlier than others, the unused computational capacity would not be allocated to other ongoing tasks. The length of a time slot is set to be a value such that all tasks can be completed during one time slot.

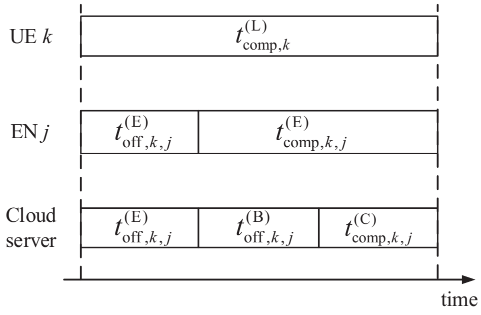

UE k
$t^{(\mathrm{L})}_{\mathrm{comp},k}$
EN j
$t^{(\mathrm{E})}_{\mathrm{off},k,j}$
$t^{(\mathrm{E})}_{\mathrm{comp},k,j}$
Cloud server
$t^{(\mathrm{E})}_{\mathrm{off},k,j}$
$t^{(\mathrm{B})}_{\mathrm{off},k,j}$
$t^{(\mathrm{C})}_{\mathrm{comp},k,j}$
time

Fig. 2. Illustration for optimal task partitioning.

can assign a certain ratio of its task to EN j only when it is associated with EN j; and constraints in (18) result from the SINR constraint as described in (6).

V. Solution Algorithms

We decompose **P1** into two levels of subproblems. The lower-level subproblem determines the optimal task partitioning ratios with a given user association. The higher-level subproblem determines the user association given that the optimal task partitioning is applied, which is solved by a dual decomposition-based approach.

A. Optimal Task Partition with Given User Association

As discussed, the latency of a task equals to the latency of subtasks that are completed at the latest among UE, EN, and cloud server. Since $\alpha_k + \sum_{j=1}^{J} x_{k,j}\beta_{k,j} + \sum_{j=1}^{J} x_{k,j}\gamma_{k,j} = 1$, a decrease of one ratio would cause an increase of at least one of the other ratios. Thus, the optimal task partitioning is achieved when the subtasks executed at the UE, the EN, and cloud server are completed at the same time, as shown in Fig. 2. Then, we have the following equations:

$$t^{(\mathrm{L})}_{k} = \sum\nolimits_{j=1}^{J} x_{k,j} t^{(\mathrm{E})}_{k,j} = \sum\nolimits_{j=1}^{J} x_{k,j} t^{(\mathrm{C})}_{k,j}. \tag{19}$$

Let α^*_k, $[\beta^*_{k,j}]_{j\in\mathcal{J}}$, and $[\gamma^*_{k,j}]_{j\in\mathcal{J}}$ be the optimal task partitioning ratios of UE k. Applying the expressions given in (9) to (19), we have:

$$\begin{aligned}\frac{\alpha^*_k s_k z_k}{c^{(\mathrm{L})}_k} &= \frac{\left(\sum\limits_{j=1}^{J}\beta^*_{k,j} + \sum\limits_{j=1}^{J}\gamma^*_{k,j}\right) s_k Q_j}{W\log\left(1+\theta_{k,j}\right)} + \frac{\sum\limits_{j=1}^{J}\beta^*_{k,j} s_k z_k Q_j}{c^{(\mathrm{E})}_j} \\ &= \frac{(\sum\limits_{j=1}^{J}\beta^*_{k,j} + \sum\limits_{j=1}^{J}\gamma^*_{k,j}) s_k Q_j}{W\log\left(1+\theta_{k,j}\right)} + \frac{\sum\limits_{j=1}^{J}\gamma^*_{k,j} s_k Q_j}{M_j} + \frac{\sum\limits_{j=1}^{J}\gamma^*_{k,j} s_k z_k}{c^{(\mathrm{C})}_k}\end{aligned} \tag{20}$$

Combine (20) with the equation $\alpha^*_k + \sum_{j=1}^{J}\beta^*_{k,j} + \sum_{j=1}^{J}\gamma^*_{k,j} = 1$, the solutions of α^*_k, $\sum_{j=1}^{J}\beta^*_{k,j}$, and $\sum_{j=1}^{J}\gamma^*_{k,j}$ can be obtained. Finally, based on $[x_{k,j}]_{j\in\mathcal{J}}$, the optimal α_k, $[\beta_{k,j}]_{j\in\mathcal{J}}$, and $[\gamma_{k,j}]_{j\in\mathcal{J}}$ are obtained.

B. Dual Decomposition-Based User Association

Let $\Gamma_{k,j}$ be the latency of UE k when associated with EN j under optimal task partitioning, which can be obtained by solving the equations given in (20). Note that it is also possible that UE k is not associated with any EN, i.e., $\sum_{j=1}^{J} x_{k,j} = 0$. For this case, the latency of UE k is its local computing time $\frac{s_k z_k}{c^{(\mathrm{L})}_k}$. Combining the two cases, the latency of UE k under the optimal task partitioning T^*_k is given by:

$$T^*_k = \sum_{j=1}^{J} x_{k,j}\Gamma_{k,j} + \left(1 - \sum_{j=1}^{J} x_{k,j}\right)\frac{s_k z_k}{c^{(\mathrm{L})}_k}. \tag{21}$$

Then, the objective function of the user association problem is given by:

$$\sum_{k=1}^{K} T^*_k = \sum_{k=1}^{K}\frac{s_k z_k}{c^{(\mathrm{L})}_k} + \sum_{k=1}^{K}\sum_{j=1}^{J} x_{k,j}\left(\Gamma_{k,j} - \frac{s_k z_k}{c^{(\mathrm{L})}_k}\right). \tag{22}$$

Let $\Delta_{k,j} \triangleq \frac{s_k z_k}{c^{(\mathrm{L})}_k} - \Gamma_{k,j}$, it can be interpreted as the achievable latency reduction of UE k when it is associated with EN j, compared to executing the task by itself. From (22), we can see that minimizing sum latency $\sum_{k=1}^{K} T^*_k$ is equivalent to maximizing the sum latency reduction $\sum_{k=1}^{K}\sum_{j=1}^{J}\Delta_{k,j}$. Then, the user association problem can be formulated as:

$$\mathbf{P2}: \max_{\{\mathbf{x}\}} \sum\nolimits_{k=1}^{K}\sum\nolimits_{j=1}^{J} x_{k,j}\Delta_{k,j} \tag{23}$$

$$\text{s.t.:}\ \sum\nolimits_{j=1}^{J} x_{k,j} \le 1,\ k\in\mathcal{K},\ j\in\mathcal{J}, \tag{24}$$

$$\sum\nolimits_{k=1}^{K} x_{k,j} \le S_j,\ j\in\mathcal{J}, \tag{25}$$

$$x_{k,j}\in\{0,1\},\ k\in\mathcal{K},\ j\in\mathcal{J}, \tag{26}$$

$$x_{k,j} = 0,\ k\in\mathcal{K},\ \forall j\notin\pi_k. \tag{27}$$

Problem **P2** is an integer programming problem that is NP-hard. To derive an effective solution algorithm, we relax the integer constraint by allowing all $x_{k,j}$ to take values in $[0,1]$. Although the relaxed problem, **P2-Relexted**, is non-convex, we apply a dual decomposition approach to obtain the solution and show that a near-optimal solution can be achieved.

A key design objective for user association is to achieve load balancing between ENs. Thus, we set $\mathbf{Q} \triangleq \{Q_1,\dots,Q_J\}$ to be auxiliary variables and add constraints $\sum_{k=1}^{K} x_{k,j} = Q_j$, $j\in\mathcal{J}$. Then, we have the following problem:

$$\mathbf{P3}: \max_{\{\mathbf{x}\}} \sum\nolimits_{k=1}^{K}\sum\nolimits_{j=1}^{J} x_{k,j}\Delta_{k,j} \tag{28}$$

$$\text{s.t.:}\ (24), (25) \text{ and } (27)$$

$$\sum\nolimits_{k=1}^{K} x_{k,j} = Q_j,\ j\in\mathcal{J}, \tag{29}$$

$$x_{k,j}\in[0,1],\ k\in\mathcal{K},\ j\in\mathcal{J}. \tag{30}$$

We apply a partial relaxation on the constraint $\sum_{k=1}^{K} x_{k,j} = Q_j$. The corresponding Lagrangian function is given by:

$$\mathcal{L}\left(\mathbf{x},\boldsymbol{\lambda}\right) = \sum_{k=1}^{K}\sum_{j=1}^{J} x_{k,j}\Delta_{k,j} + \sum_{j=1}^{J}\lambda_j\left(\sum_{k=1}^{K} x_{k,j} - Q_j\right). \tag{31}$$

Then, the dual problem of **P3** is given by:

$$\textbf{P3-Dual:}\ \min_{\{\boldsymbol{\lambda}\}} g(\boldsymbol{\lambda}) \tag{32}$$

where $\boldsymbol{\lambda}$ is the Lagrangian multiplier for the constraint (29), and $g(\boldsymbol{\lambda})$ is given by:

$$g(\boldsymbol{\lambda}) = \max_{\{\mathbf{x}\}} \mathcal{L}\left(\mathbf{x}, \boldsymbol{\lambda}\right). \tag{33}$$

The problems given in (32) and (33) are solved iteratively by UEs and ENs until convergence is achieved.

The problem described in (33) can be decomposed into K subproblems that are solved by each UE. At iteration t, UE k solves its subproblem by selecting the EN $j^{*[t]}$ that satisfies:

$$j^{*[t]} = \arg\max_{j \in \pi_k} \left\{ \Delta_{k,j}(Q_j^{[t]}) - \lambda_j^{[t]} \right\}. \tag{34}$$

After the selection, each UE sends a notice to the selected EN. Receiving the notifications from UEs, each EN j updates $\mathbf{x}_j = [x_{1,j}, \ldots, x_{K,j}]$ with the following rule:

$$x_{k,j}^{[t]} = \begin{cases} 1, & j = j^{*[t]} \\ 0, & \text{otherwise,} \end{cases} \tag{35}$$

On the other hand, the problem described in (32) can be decomposed into J subproblems that are solved by each EN separately. For each EN j, it updates $\lambda_j^{[t]}$ by:

$$\lambda_j^{[t+1]} = \lambda_j^{[t]} - \rho_j^{[t]} \eta_j^{[t]} \tag{36}$$

where $\eta_j^{[t]}$ is the gradient of $\lambda_j^{[t]}$, given by:

$$\eta_j^{[t]} = Q_j^{[t]} - \sum_{k=1}^{K} x_{k,j}^{[t]} \tag{37}$$

and $\rho_j^{[t]}$ is the step size, given by:

$$\rho_j^{[t]} = \frac{g(\boldsymbol{\lambda}^{[t]}) - g(\boldsymbol{\lambda}^{*})}{\left\| \boldsymbol{\eta}^{[t]} \right\|^2}. \tag{38}$$

After the update of $\lambda_j^{[t]}$, EN j updates $Q_j^{[t]}$ with the following:

$$Q_j^{[t+1]} = \min\{\textstyle\sum_{k=1}^{K} x_{k,j}^{[t]}, S_j\}. \tag{39}$$

Finally, EN j broadcasts the updated values of $\lambda_j^{[t]}$ and $Q_j^{[t]}$ to nearby UEs. The UEs then initiate the next iteration of EN selection. The procedure of the dual decomposition-based user association algorithm is summarized in Algorithm 1.

1) Performance Bound: To show that near-optimal solution can be achieved, we derive a lower bound on the latency performance. First, we exhaustively search all possible vectors $\mathbf{Q}$. For each $\mathbf{Q}$, we relax the feasibility constraints (26) and (27) in **P3** and solve the following linear programming (LP):

$$\mathbf{P4}: \max_{\{\mathbf{x}\}} \textstyle\sum_{k=1}^{K} \sum_{j=1}^{J} x_{k,j} \Delta_{k,j} \tag{40}$$
$$\text{s.t.:}\ (24), (25), (29) \text{ and } (30).$$

Since each element in $\mathbf{Q}$ has S_j possible values, the total number of LPs to be solved is $\prod_{j=1}^{J} S_j$. Among the $\prod_{j=1}^{J} S_j$ LPs, we find the one with the largest value of objective function $\sum_{k=1}^{K} \sum_{j=1}^{J} x_{k,j} \Delta_{k,j}$. Then, the largest value of $\sum_{k=1}^{K} \sum_{j=1}^{J} x_{k,j} \Delta_{k,j}$ is an upper bound for the sum latency reduction. Subtracting this value from the sum of UE local computing time $\sum_{k=1}^{K} \frac{s_k z_k}{c_k^{(\mathrm{L})}}$, the outcome is a lower bound for the sum latency of all UEs (i.e., performance lower bound).

Algorithm 1: Dual Decomposition-Based User Association Algorithm

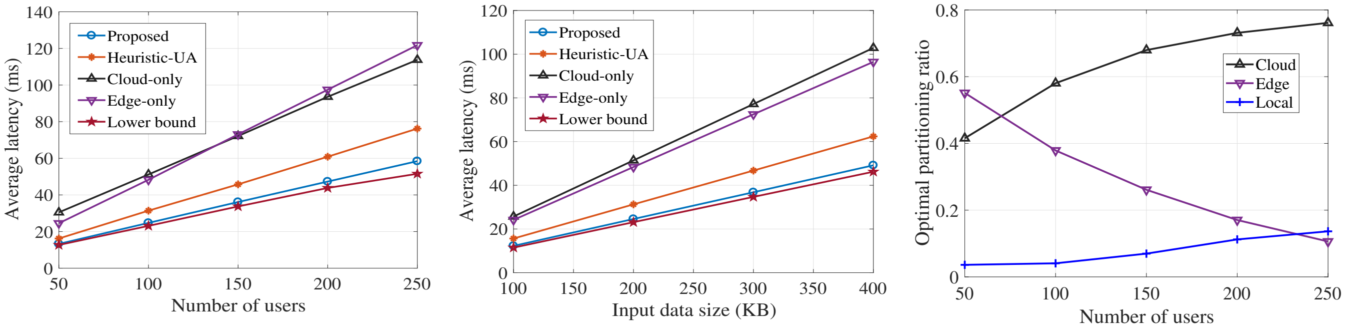
```
Initialize $\mathbf{Q}$ and $\boldsymbol{\lambda}$;
do
    for $k = 1 : K$ do
        UE $k$ selects the optimal EN according to (34) and
          informs the selected EN;
    end
    for $j = 1 : J$ do
        EN $j$ updates $\mathbf{x}_j$ according to (35);
        Updates $\eta^{[t]}$ according to (37);
        Updates $\lambda_j$ according to (36);
        Updates $Q_j$ according to (39);
    end
    $t++$
while ($\mathbf{x}$ does not converge);
```

VI. Simulation Results

We evaluate the performance of the proposed scheme with simulations. We consider a 500 m × 500 m area with 10 ENs and a varying number of users randomly located in the area. The channel model includes a distance-dependent path loss $140.7 + 36.7\log_{10}d$ in dB and Rayleigh fading, where d is the distance in meters. The UE transmission power is 20 dBm and the noise density is -174 dBm/Hz. The uplink bandwidth is 10 MHz. The data rate of each backhaul link is uniformly distributed in $[20, 80]$ Mbps. Unless otherwise stated, the default number of users is 100, the default size and complexity of the tasks are $s_k = 200$ KB and $z_k = 1000$ CPU cycles/bit, respectively. The computational capabilities of UE, EN, and the cloud server are 1 GHz, 50 GHz, and 100 GHz, respectively. Each task can be partitioned into 50 subtasks, hence the resolution for task partitioning ratios is 0.02.

The proposed scheme is compared with several benchmark schemes. The first one is *edge-only* scheme, where all subtasks are executed at the EN; the second one is *cloud-only* scheme, where all subtasks are executed at the cloud server. For a fair comparison, the dual decomposition-based user association is applied in these two schemes. The third scheme is a heuristic user association scheme (termed *heuristic-UA*), in which each UE is associated with the EN with the maximum SINR, and the optimal task partitioning is applied.

The average latency versus the number of users is shown in Figs. 3(a). We can see that the average latency reduces as the traffic load increases. When the traffic load is low, the edge-only scheme outperforms the cloud-only scheme, showing the advantage of edge computing. When the traffic load is high, the cloud-only scheme outperforms the edge-only scheme, since more users share the resources of ENs. The proposed scheme and heuristic-UA scheme outperform the previous two schemes, as the tasks are properly partitioned and assigned among UE, ENs, and cloud servers. The proposed



(a)

(b)

(c)

Fig. 3. Simulation results. (a) average latency vs. number of users, (b) average latency vs. input data size, (c) optimal partitioning ratios vs. number of users.

scheme outperforms the heuristic-UA scheme, since the set of UEs associated with each EN is optimized and a good load balancing among ENs is achieved. The performance of the proposed scheme is close to the lower bound, showing that the near-optimal user association can be achieved.

The average latency versus task input data size is presented in Fig. 3(b), where similar trends among different schemes are observed. When the data size is small, the latency reduction achieved by the proposed schemes is small, since the UE is able to execute the tasks in a timely manner. As the input data size increases, a higher latency reduction can be achieved.

The average optimal partitioning ratios versus the numbers of users is plotted in Fig. 3(c). As the number of users increases, the ratios assigned to the local device and cloud server increase, while the ratio assigned to the edge server decreases. This is because both the offloading time and computing time at the EN are higher when the traffic load increases, while the computing times at UE and cloud server are not impacted by the traffic load. To minimize the total latency, the workload assigned to the edge server should be reduced.

VII. Conclusions

In this paper, we considered the joint optimization of task partitioning and user association in MEC systems. Such a problem is formulated as a mixed-integer programming problem. We solved the formulated problem by decomposing it into two levels of subproblems and developing an efficient solution for each subproblem. Simulation results show that the proposed scheme lowers the average latency by about 50% compared to several benchmark schemes.

Acknowledgment

This research was supported in part by NSF (grants CNS-1910348, CNS-1563655, CNS-1731164, CNS-1813401, and IIP-1822071) and by the Broadband Wireless Access & Applications Center (BWAC). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF.

References

[1] Huawei, “5G Vision: 100 Billion Connections, 1 ms Latency, and 10 Gbps Throughput,” Accessed: Dec. 2020. [Online]. Available: http://support.huawei.com/huaweiconnect/carrier/en/thread-357441-1-1.html

[2] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022” White Paper, Feb. 2019. Accessed: Dec. 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html

[3] 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 Commun.,* vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[4] M. Chen and Y. Hao, “Task offloading for mobile edge computing in software defined ultra-dense network,” *IEEE J. Sel. Areas Commun.,* vol. 36, no. 3, pp. 587–597, Mar. 2018.

[5] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, “Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing,” *IEEE J. Sel. Areas Commun.,* vol. 37, no. 3, pp. 668–682, Mar. 2019.

[6] X. Chen et al., “Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning,” *IEEE Internet Things J.,* vol. 6, no. 3, pp. 4005–4018, June 2019.

[7] X. Lyu et al., “Selective offloading in mobile edge computing for the green Internet of Things,” *IEEE Netw.,* vol. 32, no. 1, pp. 54–60, Jan./Feb. 2018.

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

[9] Y. Xiao and M. Krunz, “Distributed optimization for energy-efficient fog computing in the tactile Internet,” *IEEE J. Sel. Areas Commun.,* vol. 36, no. 11, pp. 2390–2400, Nov. 2018.

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

[11] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, “Mobile-edge computing: Partial computation offloading using dynamic voltage scaling,” *IEEE Trans. Commun.,* vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[12] 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, Mar. 2017.

[13] J. Ren, G. Yu, Y. Cai, and Y. He, “Latency optimization for resource allocation in mobile-edge computation offloading,” *IEEE Trans. Wireless Commun.,* vol. 17, no. 8, pp. 5506–5518, Aug. 2018.

[14] J. Ren, G. Yu, Y. He, and G. Y. Li, “Collaborative cloud and edge computing for latency minimization,” *IEEE Trans. Veh. Technol.,* vol. 68, no. 5, pp. 5031–5044, May 2019.

[15] J. Liu and Q. Zhang, “Offloading schemes in mobile edge computing for ultra-reliable low latency communications,” *IEEE Access,* vol. 6, pp. 12825–12837, 2018.

[16] S. Sardellitti, M. Merluzzi, and S. Barbarossa, “Optimal association of mobile users to multi-access edge computing resources,” in *Proc. IEEE ICC’18,* Kansas City, MO, May 2018, pp. 1–6.

[17] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, “Joint computation offloading and user association in multi-task mobile edge computing,” *IEEE Trans. Veh. Technol.,* vol. 67, no. 12, pp. 12313–12325, Oct. 2018.

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