Online Data Driven Scheduling for Deadline-Sensitive Tasks of Mobile Edge Computing Enabled Consumer Electronics

Lichao Yang, Kailin Wang, Mingyan Xiao, Heli Zhang, Ming Li, Xi Li, Hong Ji

***Abstract*—With the remarkable prosperity in smart consumer electronics (CEs), Mobile Edge Computing (MEC) has emerged as a pivotal technology to tackle the prevalent latency issues faced by smart CEs. Consequently, this has produced multiple deadline-sensitive tasks, imposing stringent computational time requirements on the infrastructure of 6G wireless communication systems. To meet the deadline demands of tasks when offloading, research efforts have focused on improving the scheduling mode. However, a majority of modes are offline, which is unrealistic for MEC servers to schedule tasks or reserve resources according to the global information (i.e., the time and quantity of tasks released by CEs) grasped in advance. To address this concern, we propose an online data-driven scheduling mechanism maximizing the revenue of deadline-sensitive tasks in the MEC-enabled consumer electronics system. Given that the released time and deadline time are fixed, but the execution process are flexible and the execution process involves two network resources, we design a two-step online resource allocation (TORA) algorithm comprising an online spectrum allocation sub-algorithm and an online computing resource allocation sub-algorithm. Moreover, we derive a precise competitive ratio to evaluate the performance of our TORA algorithm. Finally, through extensive simulations, we demonstrate its superiority in improving system revenue.**

***Index Terms*—Mobile edge computing, online, deadline-sensitive, resource allocation, competitive ratio.**

I. Introduction

In the forthcoming sixth-generation (6G) wireless communication network, a diverse range of smart consumer electronics (CEs), such as smartphones, smart appliances, smart wearable devices, etc., lead to massive data and computing tasks with strict processing deadlines produced at the network edge [1]–[3], Mobile Edge Computing (MEC) emerges as a low-latency solution to provide computing resources close to CEs [4]–[6].

Although MEC offers remarkable advantages in reducing the processing delay of tasks, there are still numerous bottlenecks in the existing task scheduling schemes. First, the majority of studies focus on the offline scheduling mode [7]–[9], where all users information is known beforehand. For instance, [10] formulated the secure computation offloading problem to maximize users satisfaction based on grasping users requirements in advance. [11] designed an offline task offloading approach to minimize the energy consumption of all users. However, considering spatio-temporal variations in the practical networks, offline task scheduling fails to adapt to the dynamic network. Therefore, it becomes imperative to optimize network resources in an online fashion.

Subsequently, it is worth noting that previous research has investigated the online scheduling mode [12]–[15]. For example, [16] proposed a parallel offloading policy to fulfill the deadline requirements of tasks while considering the load balancing of the MEC servers. [17] addressed the average offloading cost minimization problem which took into account the maximum tolerable delay of tasks. in [18], the objective was to maximize user QoE by optimizing service selection, computation resource allocation, and task offloading decisions. Another study [19] tackled the challenge of enhancing energy efficiency and meeting desired QoE requirements through a dynamic resource allocation scheme, introducing a novel hybrid algorithm based on differential evolution and a modified first-fit heuristic. In the context of a non-orthogonal multiple access-based MEC system, [20] aimed to allocate as many app users as possible to a minimal number of edge servers using an online approach. Actually, release time and deadline would be fixed by the CEs when the tasks are released in the online scene, the MECs can flexibly schedule the tasks to ensure accomplishment before deadlines to harvest the revenue by the system [21]. However, the above researches ignored the revenue maximum problem before deadlines leading to the network resources idle or in short supply in online scene.

For proposed online algorithm, the competitive ratio to measure the performance of online algorithms is crucial, which means the ratio between the performance of the offline algorithm and the online algorithm. However, the value is difficult to obtain exact value, for instance, [22] proposed an online scheduling algorithm for file caching, which achieved the $O(\log K)$ competitive ratio. [23] introduced the Greedy-One-Restart (GOR) algorithm to minimize the offloading costs of tasks, whose competitive ratio reached $O(\sqrt{m})$. In comparison to [22], [23], we smartly use duality theory to derive a more

Manuscript received July 27, 2023; major revised November 21, 2023 and minor revised December 17, 2023; accepted January 13, 2024. This work was supported by National Natural Science Foundation of China under Grant 62171061. (*Corresponding author: Heli Zhang.*)

L. Yang is with the School of Public Health, Soochow University Medical College, Soochow, 215123, P.R. China (email:lcyang@suda.edu.cn).

K. Wang, H. Zhang, X. Li, and H. Ji, are with the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, 100876, P.R. China (email: wangkailin2017@bupt.edu.cn, zhangheli@bupt.edu.cn, lixi@bupt.edu.cn, jihong@bupt.edu.cn).

M. Xiao is with the Department of Computer Science, California State Polytechnic University, Pomona, CA, 91768, USA (email:mxiao@cpp.edu).

M. Li is with the Department of Computer Science and Engineering, The University of Texas at Arlington, TX, 76019, USA (email:ming.li@uta.edu).

precise competitive ratio to quantify the performance of our online deadline-sensitive task scheduling mechanism.

In this paper, we delve into the realm of a 6G edge cloud network and propose an online data-driven scheduling mechanism for deadline-sensitive tasks. Our research focuses on two key aspects: one is to design an online scheduling mechanism for deadline-sensitive tasks with a flexible execution sequence of tasks to maximize the revenue of fully processed tasks, and the other is to derive a precise competition ratio to demonstrate the superiority of the online algorithm. To address these objectives, we leverage dual theory principles and propose a practical two-step online resource allocation (TORA) algorithm tailored for edge networks. Moreover, we introduce a provable competitive ratio, which serves as a constant-factor benchmark to evaluate the performance of our scheduling mechanism. The derivation of this competitive ratio is supported by a comprehensive charging argument. To validate our findings, we conduct extensive simulations, affirming the exceptional performance of the TORA algorithm. In summary, the key contributions of this paper are as follows:

- *We study an online revenue maximization problem integrated with MEC.* Our focus lies in developing an online scheduling mechanism for deadline-sensitive tasks. In order to ensure that tasks are fully transmitted and computed before their deadlines and maximize the revenue of fully processed tasks, those tasks that cannot be completed by the deadline would be abandoned in time.
- *We propose the TORA algorithm, comprising an online spectrum allocation sub-algorithm and an online computing resource allocation sub-algorithm.* Diverging from other online resource allocation algorithms that primarily concentrate on a single resource, our approach involves a two-step process to jointly allocate bandwidth and computing resources, which significantly enhances the overall efficiency. Specifically, for the online spectrum resource allocation algorithm (OSRAA), we employ duality theory to propose preemption and resumption rules. As for the online computing resource allocation algorithm (OCRAA), we adopt the first-in-first-out principle.
- *We derive a provably competitive ratio for the TORA algorithm.* To measure the performance of our scheduling mechanism, we introduce the concept of a competitive ratio for the TORA algorithm. Since the TORA algorithm consists of both OSRAA and OCRAA, for OSRAA, a provably competitive ratio is obtained through a charging argument. For OCRAA, we employ geometric theory to derive a provably competitive ratio.

The remaining sections of this paper are organized as follows. In Section II, we provide a concise introduction to the system model. Section III formulates the problem of online revenue maximization. Building upon this, Section IV presents the design of a two-step online resource allocation algorithm, which encompasses the online spectrum resource allocation algorithm (OSRAA) and the online computing resource allocation algorithm (OCRAA). Furthermore, Section V presents the derivation of a competitive ratio, representing the ratio between the total revenue achieved using an optimal offline algorithm and that of our proposed online algorithm before the deadlines. In Section VI, we conduct simulations to compare the performance of our approach with existing algorithms. Finally, Section VII concludes the paper, summarizing our findings and contributions.

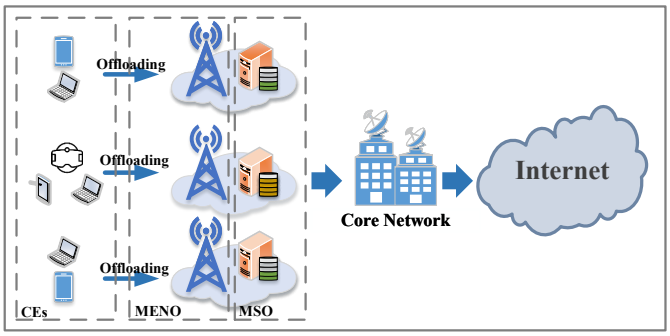


Fig. 1: System model of 6G edge cloud Network.

II. System Model

A. Network Model

We consider a practical model as illustrated in Fig.1, there are a number of MEC servers, associated with small base stations (SBSs) via the wired connection. So the nearby devices can efficiently offload computation tasks attached with deadlines to SBSs enabled with EMC servers via channels. Furthermore, each MEC server can only process a task at each time point because of limited computing resource. Thus, each SBS has a task scheduling module and a resource allocation module.

Fig. 1 consists of a set of small base stations (SBSs), denoted by $\mathcal{S} = \{1, 2, \ldots, S\}, s \in \mathcal{S}$, and each SBS is associated with one MEC server through wired manner. Then the SBSs are connected to the Internet by the core network of the MEC system. Assume that there are J CEs that are connected to SBSs by wireless links, denoted by $\mathcal{J} = \{1, 2, \ldots, J\}$. Assume that each CE has a computation task required to be executed, denoted the set of computation tasks by $X = \{x_1, x_2, \ldots, x_J\}$, $j \in J$.

We next consider two logical roles presented in Fig.1 in the edge cloud networks: mobile edge network operator (MENO) and MEC system operator (MSO). The MENO possesses the spectrum resource, while the MSO owns MEC servers. We adopt OFDMA to allocate spectrum resource in the edge cloud networks. Since the channels are orthogonal in one base station, we only consider the interference between base stations. Specially, the spectrum is divided into M bands, denoted by $\mathcal{M} = \{1, 2, \ldots, M\}$. For ease of reference, we list the key notations of our system model in Table I.

We further assume that when these tasks are released by CEs, the tasks will not be migrated to other servers among servers during execution. Moreover, assume that those tasks reach the MEC system online in arbitrary order and those tasks are released randomly over the continuous time interval $T = [0, \infty)$, as well as each MEC server can only allocate one channel to one task at a time.

TABLE I: LIST OF KEY NOTATIONS

Notation	Description
$\mathcal{S}$	Set of SBSs
$\mathcal{J}$	Set of CEs
f_{js}	The required computing resource of task x_j on server s
a_j	The arrival time of task x_j
d_j	The deadline of task x_j
$y_{js}^m(t)$	The task dispatch variable
$z_{js}^m(t)$	The server occupy variable
P_i^m	The transmission power of task x_i on channel m
η	Density of thermal noise at the receiver
g_{ij}	The channel gain power between task x_i and task x_j
γ	A constant related to the path loss
$d(i,j)$	The Euclidean distance between task x_i and task x_j
$\wedge$	A common set of two sets intersecting

B. Transmission/Interference Range and Link Capacity

Suppose the power spectral density of node i on channel m is P_i^m. Power propagation gain between node i and node j, denoted by g_{ij}, is $g_{ij} = C \cdot [d(i,j)]^{-\gamma}$, where i and j also denote the positions of node i and node j, respectively, $d(i,j)$ refers to the Euclidean distance between i and j, γ is the path loss factor, and C is a constant related to the antenna profiles of the transmitter and the receiver, wavelength, and so on. We assume that the data transmission is successful only if the received power spectral density at the receiver exceeds a threshold P_T^m. Meanwhile, we assume interference becomes non-negligible only if it produces a power spectral density over a threshold of P_I^m at the receiver[1]. Thus, the transmission range for a node i on channel b is $R_T^{i,m} = (CP_i^m/P_T^m)^{1/\gamma}$, which comes from $C(R_T^{i,m})^{-\gamma} \cdot P_i^m = P_T^m$. Similarly, based on the interference threshold $P_I^m (P_I^m < P_T^m)$, the interference range for a node is $R_I^{i,m} = (CP_i^m/P_I^m)^{1/\gamma}$, which is larger than $R_T^{i,m}$. Thus, different nodes may have different transmission ranges/interference ranges on different channels with different transmission power.

In addition, according to the Shannon-Hartley theorem, if node i sends data to node j on link (i,j) using channel m, the capacity of link (i,j) on channel m is[2]:

$$c_{ij}^m(t) = W^b(t) \log_2 \left(1 + \frac{g_{ij}(t) P_i^m}{\eta}\right), \tag{1}$$

where η is the thermal noise at the receiver. Note that the denominator inside the log function only contains η. This is because of one of our interference constraints, i.e., when node i is transmitting to node j on channel m, all the other neighbors of node j within its interference range are prohibited from using this channel.

[1]Note that the interference model we adopt in this study is the Protocol Model introduced in [24], which considers one interfering link at a time. [24] also introduces the Physical Model, according to which a transmission is successful if its signal-to-interference plus noise ratio (SINR) is above a threshold. It has been shown in [24] that these two interference models can be equivalent in terms of network capacity by setting the interference range in the Protocol Model appropriately.

[2]Note that this link capacity is the same no matter which radios the transmitter and the receiver use.

First of all, a CE cannot transmit to multiple SBSs simultaneously. Thus, we have:

$$\sum_{s \in \mathcal{S}} \sum_{m \in \mathcal{M}} y_{js}^m(t) \leq 1. \tag{2}$$

Then, a server has $\bar{M}$ idle channels at time t:

$$\sum_{j \in J} \sum_{m \in \mathcal{M}} y_{js}^m(t) \leq \bar{M}. \tag{3}$$

In addition to the above constraints at the same CE/SBS, there are also scheduling constraints due to potential interference among different transmissions. In particular, if CE i uses spectrum m to transmit data to SBS s, then any other CEs that may interfere with the reception at SBS s should not use this spectrum. In other words, other SBSs that are within CE i's interference range cannot use the same spectrum m $(\forall j,k \in J, k \neq j, s, u \in \mathcal{S}, u \neq s, u \leq R_I^{j,m}, \forall m \in \mathcal{M})$:

$$y_{js}^m(t) + y_{ku}^m(t) \leq 1. \tag{4}$$

C. Resource Allocation Constraint

Next, we illustrate the spectrum resource allocation constraint on the data transmission.

Assume that band m is available at both task x_j and MEC server j. We denote $y_{js}^m(t)$ if the task x_j is dispatched to the MEC server s by channel m at time t. $y_{js}^m(t)$ is a two dimensional binary variable, and $y_{js}^m(t) = 1$ means that the channel m is assigned to the task x_j and the MEC server s at time t. Otherwise, $y_{js}^m(t) = 0$.

$c_{js}^m(t)$ denotes the maximum link's capacity on link (j,s) via channel m at time t. We define $F_s(cycles/s)$ as the computation rate of MEC server s, and $v_m(cycles/bit)$ means the number of CPU cycles required when processing per-bit data. Denote F_s/v_m as the data processing rate threshold for channel m. In order to make full use of the spectrum and computing resources, we consider the constraint condition between the computation rate and the link's capacity. Thus, the link's capacity of the transmission process should not be more than the data processing rate on the MEC server:

$$\sum_{j \in J} \sum_{s \in \mathcal{S}} y_{js}^m(t) c_{js}^m(t) \leq F_s/v_m. \tag{5}$$

Then in order to optimize the computing resource, $z_{js}(t)$ is a binary variable, and $z_{js}(t) = 1$ denotes that the server s is occupied by task x_j at time t, otherwise $z_{js}(t) = 0$. Additionally, assume that f_{js} means the required computing resource of task x_j on server s. Thus we give the computing resource allocation constraint shown as:

$$\sum_{j \in J} z_{js}(t) f_{js} \leq F_s. \tag{6}$$

To avoid migration overhead, we do not allow the servers to migrate a task to other servers upon scheduling to a server, and one server just can execute one task at a time point:

$$\sum_{j \in J} z_{js}(t) \leq 1. \tag{7}$$

D. The Definitions

Before we introduce the formulation of the revenue maximization problem, we list some key concepts and provide descriptions of them.

Transmission Tasks Status. According to the final transmission status of tasks, we divide those tasks into two sets: fully transmitted tasks X^F, which have been completed by their deadline; partially transmitted tasks X^p, which have begun their transmission but were not completed by its deadline. In this paper, we also regard the untransmitted tasks as partially transmitted tasks set, without additional discussion.

Computation Tasks Status. As we all know, only those tasks that are transmitted successfully, will have chances to be executed on servers. Finally, according to the final computation status of tasks, we divide those tasks into two sets: Fully computed tasks $X^{F'}$ before their deadlines and $X^{F'} \in X^F$; Partially computed tasks $X^{p'}$ which have not been completed by their deadlines and $X^{p'} \in X^F$.

Revenue Gain Density. We define $\pi_{js}(t)$ as the revenue gain density of task x_j when allocated per spectrum during transmission process. Then we define $a_{js}(t)$ as the revenue gain density of task x_j when executed on MEC server s for per CPU.

Task's Release Time. Assume that at time t, there is a task x_j released from the CE, we define the time point t as r_j, named as release time of task x_j. In this paper, assume that upon one task release, the task will be connected to an MEC server instantly.

Task's Arrival Time. Suppose the arrival time of task x_j is a_j on the MEC server, i.e., the deadline of the transmission process.

Task's Deadline. Suppose the deadline of task x_j is d_j when completely executed on MEC sever.

Available Time based Task Transmission Status. In this paper, we set the total length of time for transmitting one task as $|a_j - r_j| = (\rho + 1)u_j(t)$, where $u_j(t)$ is the required extra transmission time for each task x_j at time t. To guarantee the performance of our online algorithm, we set a parameter $\rho \geq 1$. For task x_j, set $\Omega_j^{-\rho}$ for the time interval $[r_j, a_j - \rho u_j(t)]$. Synchronously, define $\Omega_j^0 = \Omega_j$ to be the available time window $[r_j, a_j]$. Correspondingly, define $A^{-\rho}(t) = \{x_j \in \mathrm{X} \mid t \in \Omega_j^{-\rho}\}$ as a set of tasks whose remaining availability time is at least ρ times their remaining transmission time at time t.

Computation Time Availability. To guarantee the tasks to be computed completely on the MEC servers, we set a minimum available time window for computation process as u_j' satisfying $u_j' \geq |d_j - a_j|$.

III. Problem Formulation

In this section, we formulate a revenue maximization problem to contain the transmission and computation process. To solve it, we divide the original revenue maximization problem into two independent subproblems, i.e., transmission revenue maximization problem and computation revenue maximization problem, respectively. First, we give the revenue maximization problem as follows:

$$
\begin{aligned}
&\max \sum_{x_j \in X} \int_{r_j}^{d_j} \sum_{m \in \mathcal{M}} \sum_{s \in \mathcal{S}} y_{js}^m(t)\pi_{js}^m(t) + \sum_{s \in \mathcal{S}} z_{js}(t)a_{js}(t)dt \\
&s.t.\ C1: \sum_{m \in \mathcal{M}} \sum_{s \in \mathcal{S}} \int_{r_j}^{d_j} y_{js}^m(t)c_{js}^m(t)dt \leq C_j, \forall m \in \mathcal{M}, s \in \mathcal{S}, \\
&\quad C2: \sum_{m \in \mathcal{M}} \sum_{s \in \mathcal{S}} y_{js}^m(t) \leq 1, \qquad \forall m \in \mathcal{M}, s \in \mathcal{S}, \\
&\quad C3: \sum_{j \in J} \sum_{m \in \mathcal{M}} y_{js}^m(t) \leq \bar{M}, \qquad \forall j \in J, m \in \mathcal{M}, \\
&\quad C4: \sum_{j \in J} \sum_{s \in \mathcal{S}} y_{js}^m(t)c_{js}^m(t) \leq F_s/v_m, \quad \forall j \in J, s \in \mathcal{S}, \\
&\quad C5: \sum_{j \in J} z_{js}(t) \leq 1, \qquad \forall j \in J, s \in \mathcal{S}, \\
&\quad C6: \sum_{j \in J} z_{js}(t)f_{js} \leq F_s, \qquad \forall j \in J, s \in \mathcal{S}, \\
&\quad C7: y_{js}^m(t), z_{js}(t) \geq 0, \qquad \forall j \in J, s \in \mathcal{S},
\end{aligned}
\tag{8}
$$

where constraint $C1$ indicates that the task j allocated MEC server s at time t can not exceed the maximized data size C_j. Constraint $C2$ means that the task j can access one MEC server through one channel at a time. Constraint $C3$ means one server has at most $\bar{M}$ available channels at time t. Then, to guarantee the efficient utilization of spectrum and computation resources, the data processing rate threshold F_s/v_m on an MEC server s can not exceed the link's capacity at time t in constraint $C4$. Constraint $C5$ shows that one MEC server can execute one task at a time point. Constraint $C6$ means that the computing resource allocated to task j at time t can not exceed the maximal capacity of server s.

Next, since the transmission process and computation process have a time execution order, we divide the original problem into the transmission revenue maximization problem and computation revenue maximization problem according to time order. Then we design two online resource allocation algorithms, to measure the performance of our proposed algorithm, the competitive ratio concept is introduced, which will be described in detail in the next section. Specially, first, we do a relaxation transformation for variable y to reformulate the optimization problem into a relaxed program problem (9). Then assume the optimal solution of problem (9) is $OPT^*(X)$ and the objective value is $e(OPT^*(X))$. Since the variable $y_{x_j,s}^m(t)$ is a discrete two-dimensional binary variable, after relaxing the variable, achieving that $e(OPT(X)) \leq e(OPT^*(X)))$. Next, utilizing the dual fitting techniques, we find the dual program problem (11). According to the weak duality, know that the objective value of the dual program problem (11) is the upper bound of problem (9). Thus, assume $OPT^{**}(X)$ is a feasible solution (which does not have to be an optimal solution) of the dual program problem, and $e(OPT^{**}(X))$ is the objective value gained by the dual program problem, indicating: $e(OPT^*(X)) \leq e(OPT^{**}(X))$.

For the convenience of description, we first formulate a transmission revenue maximization problem. In particular, we formulate a relaxed program problem (9) to maximize the transmission revenue while guaranteeing as many tasks as

possible transmitted before their arrival time:

$$\max \sum_{x_j \in X} \int_{r_j}^{d_j} \sum_{m \in \mathcal{M}} \sum_{s \in \mathcal{S}} y_{js}^m(t) \pi_{js}^m(t) dt \quad s.t.\ C1, C2, C3, C4, C7, \quad \forall j \in J, s \in \mathcal{S}. \tag{9}$$

Then, we formulate a computation revenue maximization problem with a relaxed program, given by:

$$\max \sum_{x_j \in X} \int_{r_j}^{d_j} \sum_{s \in \mathcal{S}} z_{js}(t) a_{js}(t) dt \quad s.t.\ C5, C6, C7, \qquad \forall j \in J, s \in \mathcal{S}. \tag{10}$$

IV. Two-Step Online Resource Allocation Algorithm

In this section, we focus on the design of a two-step online resource allocation (TORA) algorithm consisting of an online spectrum resource allocation algorithm (OSRAA) and an online computation resource allocation algorithm (OCRAA). More details will be listed as follows:

A. Online Spectrum Resource Allocation Algorithm

To design OSRAA, we first adopt the dual fitting approach to transform the relaxed program problem (9) into the following dual program problem (11). By solving the following dual program problem (11), the OSRAA finds the optimal solution to the relaxed program problem (9):

$$\begin{aligned} \min & \sum_{j \in J} \int_0^{\infty} C_j \gamma_j(t) dt + \sum_{j \in J} \int_0^{\infty} \zeta_j(t) dt + \\ & \sum_{s \in S} \int_0^{\infty} \bar{M} \epsilon_s(t) dt + \sum_{m \in \mathcal{M}} \int_0^{\infty} F_s / v_m \kappa_m(t) dt \\ s.t.\ & C8: c_{js}^m(t) \gamma_j(t) + \zeta_j(t) + \epsilon_s(t) + c_{js}^m(t) \kappa_m(t) \geq \pi_{js}^m(t), \\ & C9: \gamma_j(t), \zeta_j(t), \epsilon_s(t), \kappa_m(t) \geq 0. \end{aligned} \tag{11}$$

For the dual program problem (11), we introduce four dual variables $\gamma_j(t), \zeta_j(t), \epsilon_s(t)$ and $\kappa_m(t)$ for the constraints $C1, C2, C3$ and $C4$ in relaxed program problem 9. In order to find a set of feasible solution, we do the following analysis: Specially, to ensure the task full transmission, we set variable $\gamma_j(t) = \pi_{js}^m(t)/c_{js}^m(t)$ at time t. Second, we set the variable $\zeta_j(t) = 0$ which ensures the constraint $C2$ in the relaxed program problem. Then, it is obvious that the variable $\epsilon_s(t)$ is related to the allocation of servers. In this paper, we just consider the transmission process, so these dual variables $\epsilon_s(t)$ will be set to 0. Finally, the variable $\kappa_m(t)$ is dependent on the partially transmitted tasks, in the analysis for partial tasks, we introduce a continuous function $\kappa_m(t) : R^+ \rightarrow R^+$ one per channel.

From the above analysis, note that our goal is to construct a feasible solution to the dual program problem that could cover dual constraint $C8$. Notice that we set $\gamma_j(t) = \pi_{js}^m(t)/c_{js}^m(t)$ for each completely transmitted task $x_j \in X^F$. Since the meaning of the expression is the revenue per bit of task x_j, this step increases the dual revenue by exactly $\sum_{j \in X} \int_0^{\infty} C_j \gamma_j(t) dt = \sum_{j \in X} \int_0^{\infty} C_j \pi_{js}^m(t)/c_{js}^m(t) dt = v(X^F)$. To cover the remaining dual constraints of partially transmitted tasks, we introduce the $\kappa_m(t)$ function corresponding time t. So to obtain a feasible solution to the dual program problem, we require function $\kappa_m(t)$ to satisfy for every time $t \in R^+ : \kappa_m(t) \geq max\{\pi_{js}^m(t)/c_{js}^m(t) | x_j \in A(t) \wedge x_j \in X^p\}$.

Algorithm 1 Online Spectrum Resource Allocation Algorithm (OSRAA)

Input: assume there is a released task x_j with release time r_j and arrival time a_j, a transmitting task x_i on channel m with data transmission rate v_m, total channel set $\mathcal{M}$, excluded channel set M^0, available channel set M', rejected tasks set X^0
Initialize: $c_{js}^m(t)$, $\pi_{js}^m(t)$ and F_s
while $t = r_j$ **do**
  task x_j chooses the MEC server s randomly;
  for $m \in \mathcal{M}$ **do**
    if $s \in R_I^{i,m}$ and $c_{js}^m \leq F_s / v_m$ **then**
      $m \in M^0$
    end if
  end for
  $M' = \mathcal{M} - M^0$;
  if $m \in M'$ **then**
    derive the profile $\iota_{js}^m(t) = \{\pi_{js}^m(t)/c_{js}^m(t)\}$ of each task x_j;
    rank $\iota_{js}^m(t)$ for all the channels of MEC server s in descending order;
    task x_j chooses one most preferred channel m from M';
    Preemption Rule:
    if $\iota_{js}^m(t) \geq \iota_{is}^m(t)$ **then**
      a released task x_j preempts a transmitting task x_i;
      $y_{js}^m(t) = 1$;
    else if $\iota_{js}^m(t) < \iota_{is}^m(t)$ **then**
      task x_j is rejected by MEC server s;
      $x_j \in X^0$;
    end if
    Resumption Rule:
    if $t \leq a_j - \rho u_j(t)$ **then**
      task x_j will send request to suboptimal channel m' according to profile $\iota_{js}^{m'}(t) = \{\pi_{js}^m(t)/c_{js}^m(t)\}$;
      repeat step 5-22;
      $y_{js}^m(t) = 1$;
    else if $a_j - \rho u_j(t) < t < a_j$ **then**
      task x_j will not be resumption any more;
      $y_{js}^m(t) = 0$;
    end if
  end if
end while
Output: $y = \{y_{js}^m(t)\}$.

At first, we aim to design an OSRAA based on this function $\kappa_m(t)$ related to the spectrum allocation during the transmission process. However, notice that if we use variable $\kappa_m(t)$ to design the OSRAA, some challenges will arise. Since we introduce the definition of slackness, the design of function $\kappa_m(t)$ can not cover all the schedule of partially transmitted tasks. Accordingly, we design a new function $\kappa_m^{-\rho}(t) : R^+ \rightarrow R^+$, defined by:

$$\kappa_m^{-\rho}(t) = \max_{x_j} \{\pi_{js}^m(t)/c_{js}^m(t) \mid x_j \in A^{-\rho}(t) \wedge x_j \in X^p\}. \tag{12}$$

The function $\kappa_m^{-\rho}(t)$ has one appealing property: When x_j satisfies $x_j \in A^{-\rho}(t)$, the function $\kappa_m^{-\rho}(t)$ could cover the constraints of the dual program problem. This nearly completes the analysis. Then we use $\kappa_m^{-\rho}(t)$ and parameter ρ to design OSRAA. Specially, once the task is released, it will

Algorithm 2 Online Computing Resource Allocation Algorithm (OCRAA)

Input: The original scheduling sequence $Q_{schedule}$, newly dispatched task j with a minimum available time window for computation process u'_j.
Output: Completely executed task sequence Q_{best}.
Initialize: A series of completely transmitted tasks X^F and $z_{js}(t)$;
$Q_{best} \leftarrow \emptyset$;
repeat
  while $a_j \leq t \leq d_j - u'_j$ **do**
    do a descending order according to the revenue gain density of tasks
    for $j \in \max\{z_{js}(t)\}$ **do**
      $Q'_{schedule} \leftarrow$insert j into original scheduling sequence $Q_{schedule}$;
    end for
  end while
until task j is completely executed;
$Q_{best} \leftarrow j$;
Output: Q_{best};

first choose one MEC server randomly. Then the CE j applies the OSRAA to choose one channel m satisfying formula 12 on server s. Once the channel m is occupied by the task x_j, when other tasks arrive in the same channel, it will use OSRAA to pick one task that owns the largest value of $\kappa_m^{-\rho}(t)$. Finally, the main procedures of OSRAA are listed in Algorithm 1. Then to guarantee the tasks transmitted completely as many as possible, as well as to meet online features of scheduling, we add two rules in the OSRAA, i.e., preemption and resumption rules, which are highlighted in the following section.

Preemption Rule: A released task x'_j can preempt a transmitting task x_j on channel m at time t, when $\pi^m_{j's}(t)/c^m_{j's}(t) > \pi^m_{js}(t)/c^m_{js}(t)$.

The preemption rule is primarily designed to determine the task that should be transmitted upon its release. Specifically, if a newly released task x'_j has a higher value than the current task x_j on channel m, the preemption rule dictates that the new task x'_j will preempt the current task x_j to be transmitted through channel m. However, solely relying on the preemption rule has a drawback: it allows a smaller task to preempt a larger task with significantly higher revenue. This can result in a substantial loss of revenue. To mitigate this issue, we leverage the resumption rule to compensate for the lost revenue.

Resumption Rule: If at some time t, some tasks are preempted or not transmitted yet, but their remaining available time is larger than $u_j(t)$, they will consider choosing one new channel.

The resumption rule is applicable in two distinct scenarios. Firstly, for a task x_j that remains untransmitted during the time interval $[r_j, a_j - \rho u_j(t)]$ at time t, it will not be transmitted thereafter. However, if another task x'_j intends to transmit before the untransmitted task x_j, it must possess a revenue higher than $\pi^m_{js}(t)/c^m_{js}(t)$. This criterion ensures that the preempting task has a higher potential for revenue generation. Secondly, if a task x_j has been partially transmitted (but remains incomplete), any other tasks transmitted during the time interval $[a_j - \rho u_j(t), a_j]$ must have a revenue of at least $\pi^m_{js}(t)/c^m_{js}(t)$. This condition guarantees that the revenue generated by the concurrently transmitting tasks is at least proportional to their allocated resources, ensuring fairness and efficiency in the allocation process.

B. Online Computing Resource Allocation Algorithm

Upon completing the Online Spectrum Resource Allocation Algorithm (OSRAA) during the transmission process, we proceed to design the Online Computing Resource Allocation Algorithm (OCRAA) for MEC servers. The OCRAA algorithm is presented in detail in Algorithm 2. Specifically, when tasks arrive at the MEC servers, the servers first update the $Q'_{schedule}$ by arranging the tasks in descending order based on their revenue gain density. Then, the algorithm selects the task x_j with the highest revenue gain density and adds it to the sequence of completely executed tasks, Q_{best}. Once task x_j is completed before its deadline, the server performs a new scheduling process that takes into account the current time. This ensures that the scheduling remains adaptive and responsive to the changing task dynamics.

V. Analysis of Online Scheduling Algorithm-Competitive Ratio

In this section, our primary focus is to evaluate the performance of our proposed online data driven scheduling mechanism using the concept of a competitive ratio. To begin, we provide the following definition of the competitive ratio.

Competitive Ratio. Let X represent a series of tasks. We denote $e(TORA(X))$ as the total revenue achieved through the TORA algorithm, while $e(OPT(X))$ refers to the total revenue obtained from the objective value of an optimal offline allocation problem (i.e., the objective value of the relaxed program problem). To ensure and assess the worst-case performance of the TORA algorithm, we define a maximal specific value o as the competitive ratio. This ratio is calculated as the revenue of tasks completed before the deadlines using the optimal algorithm, divided by the revenue obtained through our online algorithm, which is expressed as $e(OPT(X))/e(TORA(X))$. Below, we present the expression for the competitive ratio:

$$o = \max_X \{\frac{e(OPT(X))}{e(TORA(X))}\}. \tag{13}$$

From the above expression, we say the TORA algorithm is $o-$competitive for some $o \geq 1$, and it achieves at least $1/o$ of the optimal offline revenue value in the worst case. In addition, the smaller the competitive ratio, the better the performance of the TORA algorithm. Thus, the main focus and challenge of this section are to compute the minimal constant value o, and we list the detailed computing procedures below.

Notice that for the TORA algorithm, which contains OSRAA and OCRAA, the competitive ratios of those two online algorithms can be derived respectively. Only the fully transmitted and computed tasks can gain revenue, and the partially transmitted and computed tasks do not result in partial revenue. Thus, we set $e(X^F)$ as the transmission revenue increased by the fully transmitted tasks according to OSRAA. Then we

set $e(X^{F'})$ as the computation revenue increased by the fully computed tasks according to OCRAA. Hence, o_1 denotes the competitive ratio of the OSRAA and o_2 is the competitive ratio of OCRAA. In this section, we first give the proof of o_1, so we need to find the upper bound of $e(OPT(X))$ which is related to $e(X^F)$, set as $e(OPT_1(X))$. In this paper, since $e(OPT_1(X))$ can not be obtained easily, we can derive its upper bound $e(OPT_1^{**}(X))$ by solving the dual program problem (11). Furthermore, we introduce the proof of o_2, and set $e(OPT(X))$ related to $e(X^{F'})$ as $e(OPT_2(X))$ by solving the problem (10).

A. Competitive Ratio of OSRAA

However, how should we obtain the revenue value of dual program problem 11? We will list the detailed analysis process in the following section. Specially, notice that the first part of the objective function is $\sum_{x_j \in X^F} \int_0^\infty C_j \gamma_j(t)dt = \sum_{x_j \in X^F} \int_0^\infty C_j \pi_{js}^m(t)/c_{js}^m(t) = e(X^F)$. Thus, we just focus on the analysis of the second part of the objective function to derive the upper bound of the objective value of problem 11 by the following lemma1.

LEMMA 1. Let function $\kappa_m(t)$ be satisfying: $\kappa_m(t) \geq \pi_{js}^m(t)/c_{js}^m(t)$ for every task $x_j \in X^p$ when $t \in \Omega_j$. Then there exists function $\kappa_m^{-\rho}(t)$ satisfying: $\kappa_m^{-\rho}(t) \geq \pi_{js}^m(t)/c_{js}^m(t)$ for every task $x_j \in X^p$ when $t \in \Omega_j^{-\rho}$, such that:

$$\sum_{m \in \mathrm{M}} \int_o^\infty \kappa_m(t)dt \leq \sum_{m \in \mathrm{M}} (1+\rho) \int_0^\infty \kappa_m^{-\rho}(t)dt. \tag{14}$$

The geometrical relationship comes from [25]. The proof of Lemma 1 is shown in APPENDIX. According to simulations in Section VI, we find that the parameter ρ has a great effect on the geometric formula 14.

When $t \in \Omega_j$, we set $\gamma_j(t) = \pi_{js}^m(t)/c_{js}^m(t)$ for every fully transmitted task $x_j \in X^F$, and $\gamma_j(t) = 0$ otherwise. In addition, to cover the remaining dual constraints, we apply Lemma 1 on the function $\kappa_m(t)$ and $\kappa_m^{-\rho}(t)$. Then the variable $\iota_s(t)$ and $\epsilon_s(t)$ are all set to 0. Thus the dual revenue at most:

$$\begin{aligned}
&\sum_{j \in\ X} \int_0^\infty C_j \gamma_j(t)dt + \sum_{m \in \mathcal{M}} \int_0^\infty F_s/v_m \kappa_m^{-\rho}(t)dt + \\
&\leq e(\mathrm{X}^F) + \sum_{m \in \mathrm{M}} (1+\rho) F_s/v_m \int_0^\infty \kappa_m^{-\rho}(t)dt \\
&\leq e(\mathrm{X}^F) + \sum_{m \in \mathrm{M}} (1+\rho) F_s/v_m \int_0^\infty \bar{F}_m(t)dt,
\end{aligned} \tag{15}$$

where function $\bar{F}_m(t)$ represents the gained revenue-density of the task transmitted by OSRAA at time t. Thus, $\int_0^\infty \bar{F}_m(t)$ represents the revenue gained by the OSRAA at every time t. To derive the competitive ratio, we next need to bound $\int_0^\infty \bar{F}_m(t)dt$ by applying a charging argument.

A Charging Argument

THEOREM 1. Let X^F represent the set of tasks that are fully transmitted by the OSRAA given the input tasks X. We denote $e(X^F)$ as the total revenue value obtained by this algorithm. Additionally, let $\bar{F}_m(t)$ denote the revenue density achieved by OSRAA for the task transmitted at time t on channel m. Based on these definitions, we can establish the following inequality:

$$\sum_{m \in \mathrm{M}} \int_0^\infty \bar{F}_m(t)dt \leq \frac{(\rho-1)(1-\tau)}{\rho - \tau\rho - 1} \cdot e(\mathrm{X}^F). \tag{16}$$

How do we get the Theorem 1? First, we group the whole timeline into two parts based on whether the task is completed or not: T^F is a total time interval representing that all the tasks will be fully transmitted eventually, and T^p is another time interval representing that tasks will only be partially transmitted, satisfying $T^F \cup T^p = \emptyset$. Thus, we introduce the Lemma 2.

LEMMA 2. For every time $t \in T^F$, assume that $t_j \in T^F$ is the time of task x_j fully transmitted. We set $\Phi(t_j) = \bar{F}_m(t)$ for every time t. $\Phi'(t_j)$ is the final revenue of $\Phi(t_j)$ at the end of the procedure. Then we introduce a probability parameter τ satisfies: $0 < \tau < \frac{\rho-1}{\rho}$(we will give detailed explanation in the next section), Finally, a relationship between final charge function $\Phi'(t_j)$ and charge function $\Phi(t_j)$ is derived by:

$$\Phi'(t_j) \leq \frac{(\rho-1)(1-\tau)}{\rho - \tau\rho - 1} \cdot \Phi(t_j). \tag{17}$$

In order to prove Lemma 2, we will give the proof from the following two steps: the pricing process and the price transfer process respectively. First, we give the way of pricing. Notice that $sd_j = [s_j, a_j]$ as the interval between the task admission time and its deadline, which is at least $\rho u_j(s_j)$. Thus, the total time in sd_j during which OSRAA scheduled tasks different than task x_j is at least $(\rho-1)u_j(s_j)$. We use $\frac{u_j(s_j)}{(\rho-1)u_j(s_j)} = \frac{1}{\rho-1}$ as the price factor of partially processed task x_j for every transfer. The price factor is also appropriate for the time interval sd_j.

Second, we give the price transfer process: Since partially transmitted tasks do not gain revenue for OSRAA, by transferring the charge for partially transmitted tasks to that for fully transmitted tasks to obtain the final charge function $\Phi'(t)$, which contains two parts, one is the revenue gained by fully transmitted tasks, the other is that transferred by partially transmitted tasks. Then we will take the transfer process of one partially transmitted task as an example to give the detailed charge argument. Specially, assume a partially transmitted task x_j at time t_j, transferred to a fully transmitted task x'_j at time $t_{j'}$ by multiple transfers. After finishing the transfer process, for partially transmitted task x_j, $\Phi'(t_j) = 0$ for $t_j \in T^p$; for fully processed task x'_j, $\Phi'(t_{j'}) = \Phi(t_{j'}) + \Delta\Phi(t_j)$ for $t_{j'} \in T^F$, where $\Delta\Phi(t_j)$ is an increment function of charging from task x_j to task x'_j by k transfers. Specially, $\Delta\Phi(t_j) = \tau^i(\frac{1}{\rho-1})^k \Phi(t_{j'})$, where τ^i is a probability parameter for the ith partially transmitted task selected. Finally, we achieve revenue $\int_0^\infty \Phi'(t_{j'})dt$ by operating the charging procedure, which satisfies the following inequation $\sum_{m \in \mathrm{M}} \int_0^\infty \bar{F}_m(t)dt = \sum_{m \in \mathrm{M}} \int_0^\infty \Phi'(t_{j'})dt$.

Assume there exist i partially transmission tasks which don't contain task x_0. After k transfers, the charge for partially transmitted task x_0 is transferred to the fully transmitted task x'_j. We will remake a charge $\Phi'(t_{j'})$ for task x'_j: Know that

the number of transfer is C_i^{k-1}, and let $t_0 \to t_1 \to t_2 \to ...t_{k-1} \to t_{j'}$ denote the path of task x_0 transferred to task x'_j. So the final charge for task x'_j is given as follows:

$$\begin{aligned}
&\Phi'(t_{j'}) \\
&= \Phi(t_{j'}) + \Delta\Phi(t_j) \\
&= \Phi(t_{j'}) + \tau^{i+1}\sum_{i=0}^{\infty}\sum_{k=1}^{i+1} C_i^{k-1}(\frac{1}{\rho-1})^k\Phi(t_{j'}) \\
&= \Phi(t_{j'}) + \tau^{i+1}\sum_{i=0}^{\infty}\sum_{k=1}^{i+1} C_i^{k-1}(\frac{1}{\rho-1})^k\Phi(t_{j'}) \\
&= \Phi(t_{j'}) + \frac{\tau^{i+1}}{\rho-1}\sum_{i=0}^{\infty}\sum_{k=1}^{i+1} C_i^{k-1}(\frac{1}{\rho-1})^{k-1}\cdot 1^{i-k+1}\cdot\Phi(t_{j'}) \\
&= \Phi(t_{j'}) + \frac{\tau^{i+1}}{\rho-1}\sum_{i=0}^{\infty}(1+\frac{1}{\rho-1})^i\Phi(t_{j'}) \\
&= \Phi(t_{j'}) + \frac{\tau}{\rho-1}\lim_{n\to\infty}\sum_{i=0}^{n}(\frac{\tau\rho}{\rho-1})^i\Phi(t_{j'}) \\
&= \Phi(t_{j'}) + \frac{\tau}{\rho-1}\lim_{n\to\infty}\{\frac{1\cdot(1-(\frac{\tau\rho}{\rho-1})^n)}{1-\frac{\tau\rho}{\rho-1}}\}\Phi(t_{j'}) \\
&= \Phi(t_{j'}) + \frac{\tau}{\rho-1}\cdot\frac{1}{1-\frac{\tau\rho}{\rho-1}}\cdot\lim_{n\to\infty}\{1-(\frac{\tau\rho}{\rho-1})^n\}\Phi(t_{j'}) \\
&= \Phi(t_{j'}) + \frac{\tau}{\rho-\tau\rho-1}\cdot\{1-\lim_{n\to\infty}(\frac{\tau\rho}{\rho-1})^n\}\Phi(t_{j'}).
\end{aligned} \tag{18}$$

In formula (18), from the fourth equation to the fifth equation, the binomial form is used, i.e., $\sum_{k=1}^{i+1} C_i^{k-1}(\frac{1}{\rho-1})^{k-1}1^{i-k+1} = (1+\frac{1}{\rho-1})^i$. In addition, it can be observed that $(\frac{\tau\rho}{\rho-1})^i$ is an geometric series with a common ratio of $\frac{\tau\rho}{\rho-1}$ and an initial value of 1, which can be obtained by the summation formula of geometric series, i.e., $\sum_{i=0}^{n}(\frac{\tau\rho}{\rho-1})^i = (\frac{\tau\rho}{\rho-1})^0 + ... + (\frac{\tau\rho}{\rho-1})^n = \{\frac{1\cdot(1-(\frac{\tau\rho}{\rho-1})^n)}{1-\frac{\tau\rho}{\rho-1}}\}$. To bound formula (18), set probability parameter satisfying the followings:

$$\Phi'(t_{j'}) \le \begin{cases} \frac{(\rho-1)(1-\tau)}{\rho-\tau\rho-1}\cdot\Phi(t_{j'}), & 0<\tau<\frac{\rho-1}{\rho}; \\ \infty, & \frac{\rho-1}{\rho}\le\tau\le 1. \end{cases} \tag{19}$$

To bound $\int_0^{\infty} F_m(t)dt$, parameter τ needs to meet the condition $0 < \tau < \frac{\rho-1}{\rho}$. Hence, the proof of Lemma 2 is completed.

Next, we can take the integral of both sides of the inequation (19), achieving that $\int_0^{\infty}\Phi'(t_{j'})dt \le \frac{(\rho-1)(1-\tau)}{\rho-\tau\rho-1}\cdot\int_0^{\infty}\Phi(t_{j'})dt$. By simplifying, we have:

$$\begin{aligned}
\sum_{m\in\mathrm{M}}\int_0^{\infty}\bar{F}_m(t)dt &= \sum_{m\in\mathrm{M}}\int_0^{\infty}\Phi'(t)dt \\
&= \sum_{m\in\mathrm{M}}\int_{t\in T^p}\Phi'(t)dt + \sum_{m\in\mathrm{M}}\int_{t\in T^F}\Phi'(t)dt \\
&= \sum_{m\in\mathrm{M}}\int_{t\in T^F}\Phi'(t)dt \\
&\le \frac{(\rho-1)(1-\tau)}{\rho-\tau\rho-1}\cdot\sum_{m\in\mathrm{M}}\int_{t\in T^F}\Phi(t_{j'})dt \\
&= \frac{(\rho-1)(1-\tau)}{\rho-\tau\rho-1}\cdot e(X^F).
\end{aligned} \tag{20}$$

To sum up, the proof of the theorem 1 is finished.

COROLLARY 1. The competitive ratio of OSRAA is at most:

$$o_1 \le 1 + \frac{F_s/v_m(1+\rho)(\rho-1)(1-\tau)}{(\rho-\tau\rho-1)}. \tag{21}$$

Proof: In terms of problem formulation analysis, we learn that there exists the following relationship for the revenue among online transmission revenue maximization problem, linear program problem and dual program problem, i.e., $e(OPT_1(\mathrm{X})) \le e(OPT_1^*(\mathrm{X})) \le e(OPT_1^{**}(\mathrm{X}))$. We have:

$$\begin{aligned}
o_1 &= \max_{\mathrm{X}}\{\frac{e(OPT_1(\mathrm{X}))}{e(OSRAA(\mathrm{X}))}\} \\
&\le \frac{e(OPT_1^*(\mathrm{X}))}{e(X^F)} \\
&\le \frac{e(OPT_1^{**}(\mathrm{X}))}{e(X^F)} \\
&\le \frac{e(X^F)+e(X^F)\cdot\{\frac{(1+\rho)(\rho-1)(1-\tau)}{\rho-\tau\rho-1}\}}{e(X^F)} \\
&= 1 + \frac{F_s/v_m(1+\rho)(\rho-1)(1-\tau)}{(\rho-\tau\rho-1)}.
\end{aligned} \tag{22}$$

B. Competitive Ratio of OCRAA

From the theoretical analysis of this section, the competitive ratio of OCRAA is analyzed in this section.

THEOREM 2. A no preemptive online OCRAA has a competitive ratio of $\max\{\frac{F_s(d_j-a_j)}{c_j v_j}\}$.

Proof: The proof of Theorem 2 bears similarities to Lemma 1. The distinction lies in the fact that the offline optimal algorithm possesses information regarding the arrival time and deadline of all tasks before scheduling each task on the MEC servers. Conversely, the OCRAA dynamically schedules tasks with varying time constraints but lacks knowledge of future task arrivals. Consequently, the revenue generated by the offline algorithm surpasses that of the online algorithm, primarily due to the emphasis on the time component $u'_j = \frac{C_j v_j}{F_s}$. By considering the characteristics of both online and offline algorithms, we establish the following geometric relationship:

LEMMA 3. Let function $Z_{js}(t)$ be satisfying: $Z_{js}(t) \ge z_{js}(t)$ for every task when $t \in |d_j - u'_j|$ in a offline fashion. Then there exists function $Z'_{js}(t)$ satisfying: $Z'_{js}(t) \ge z_{js}(t)$ for every task when $t \in |d_j - a_j|$ in an online fashion, such that:

$$\begin{aligned}
\int_0^{\infty} Z_{js}(t)dt &\le \frac{d_j-a_j}{u'_j}\int_0^{\infty} Z'_{js}(t)dt \\
&\le \frac{d_j-a_j}{\frac{C_j v_j}{F_s}}\int_0^{\infty} Z'_{js}(t)dt \\
&\le \frac{F_s(d_j-a_j)}{C_j v_j}\int_0^{\infty} Z'_{js}(t)dt.
\end{aligned} \tag{23}$$

This article has been accepted for publication in IEEE Transactions on Consumer Electronics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCE.2024.3362350

Thus, the competitive ration o_2 can be derived as follows:

$$\begin{aligned} o_2 &= \max_{\mathrm{X}}\{\frac{e(OPT_2(\mathrm{X}))}{e(OCRAA(\mathrm{X}))}\} \\ &\leq \frac{\int_0^\infty Z_{js}(t)dt}{\int_0^\infty Z'_{js}(t)dt} \\ &\leq \frac{F_s(d_j - a_j)}{C_j v_j}. \end{aligned} \tag{24}$$

In summary, according to Theorem 2 and Lemma 3, the competitive ratio of OCRAA is at least $\max\{\frac{F_s(d_j-a_j)}{c_j v_j}\}$.

VI. Simulation Results

comparing with other existing online algorithms, called online scheduling algorithm [26] and $\alpha-$fairness algorithm [27], respectively. We also compare TORA algorithm with the partial TORA algorithm, which is only the task preemption process without resumption process by adjusting TORA algorithm without parameter ρ.

A. Simulations Settings

We consider a scenario with 20 tasks and 5 MEC servers which covers $100m \times 100m$ area [28]. the bandwidth between CE and MEC-SBS is 9MHz [29]. The transmission power ranges from $1W$ to $2W$ randomly [8], and the environment Gaussian noise is $\sigma_s = -100dBm$ [29]. According to the wireless channel model for cellular radio environment, we set channel gain $g = K \cdot [d]^{-o}$, where $K = 62.5$, $o = 4$ and d is the distance between CEs and MEC servers [30]. The computation resource of each MEC server is characterized by the number of CPU cycles per second, which is randomly distributed from 6×10^9cycle/s to 9×10^9cycle/s [29]. Each task is characterized by the input data size, which is randomly released from 4MB to 6MB [28], and the number of required CPU cycles for each MB is $\eta = 1000cycle/MB$. The revenue for fully processed tasks is randomly distributed from 50 to 60.

B. Simulations Results

Figure 2 presents the impact of the number of MEC servers on revenue. We consider the number of MEC servers ranging from 5 to 25 while keeping the number of tasks fixed at 20. Four algorithms are evaluated: the TORA algorithm, the partial TORA algorithm, the α-Fairness algorithm, and the online scheduling algorithm. In Figure 2, it can be observed that as the number of servers increases, all algorithms converge to a constant optimal revenue. This occurs because, despite the increase in the number of servers, the total revenue remains unaffected since the number of tasks is fixed. Additionally, it is notable that all algorithms achieve the same revenue when the number of MEC servers exceeds 20. This is due to the fact that, in such cases, each server on average processes only one task. Consequently, all tasks can be fully transmitted and executed, regardless of the algorithm employed.

Fig.3 demonstrates the revenue affected by the number of tasks, whose number is changing from 20 to 60 in the simulations. It can be seen that for the TORA algorithm we

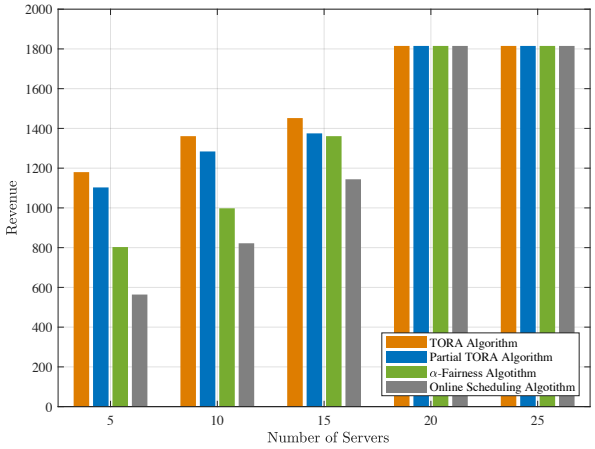


Fig. 2: Revenue versus the number of servers.

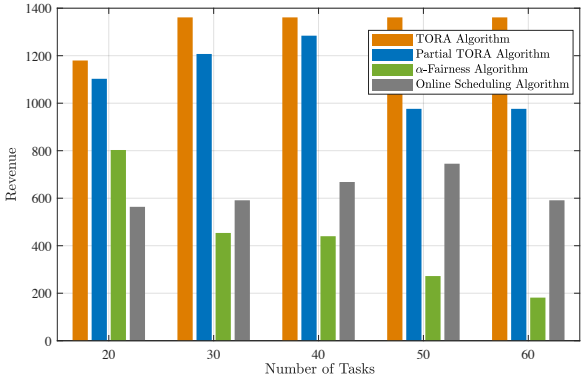


Fig. 3: Revenue versus the number of tasks.

proposed, 5 MEC servers can bear 30 tasks at most. For the partial TORA algorithm, since the tasks that are preempted cannot be resumed, when the number of tasks exceeds 40, the frequent preemption of tasks causes a decrease in the revenue. For the $\alpha-$Fairness algorithm, with the number of tasks increasing, the revenue decreases. That is because when the tasks are released, they will divide the spectrum and computing resources equally, which takes a long time to finish processing one task and a great many tasks cannot be completely executed before their deadline, so their revenue cannot be harvested by the system. In this case, with more tasks, the efficiency and the revenue of the system is lower. For the online scheduling algorithm, it can be seen that the revenue grows slowly with the number of tasks increasing, but its revenue is the smallest among the four algorithms. When the number of tasks reaches 60, the system cannot bear it, so the revenue begins to decrease.

In Fig.4 analyzes the convergence of those four algorithms. In this experiment, we set 5 MEC servers and 20 tasks and evaluate the change of the revenue with the number of iterations. It is observed that the online scheduling algorithm reaches the minimal revenue, the revenue of the partial TORA algorithm and that of $\alpha-$Fairness algorithm are ranked second and third respectively, and the TORA algorithm achieves the maximal revenue and it owns the fast convergent rate, meanwhile, saves the computation time, which can be seen

This article has been accepted for publication in IEEE Transactions on Consumer Electronics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCE.2024.3362350

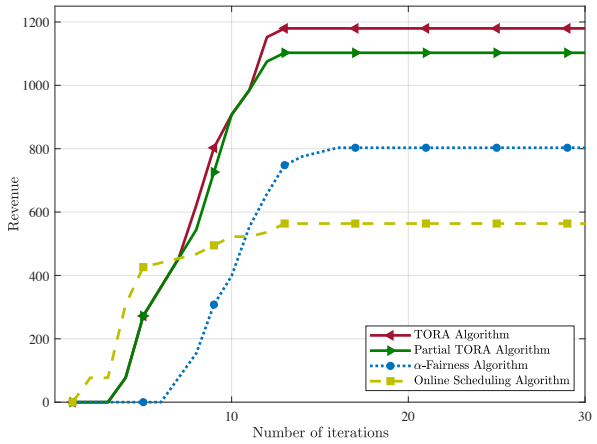


Fig. 4: Revenue versus the number of iterations under different algorithms.

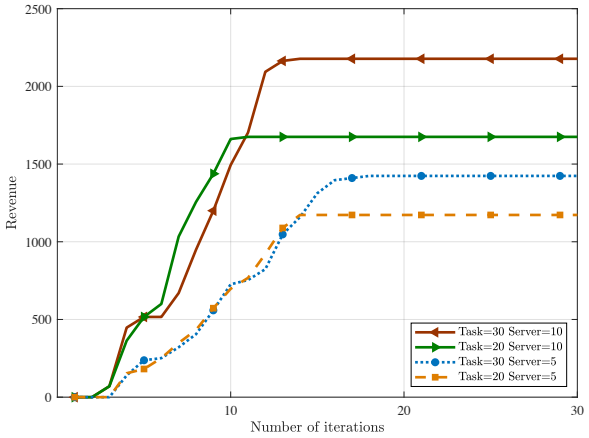


Fig. 5: Revenue versus the number of iterations under different number of tasks and servers.

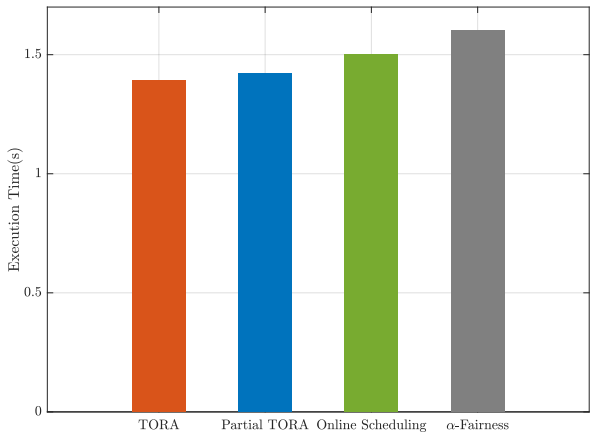


Fig. 6: The execution time of different algorithms.

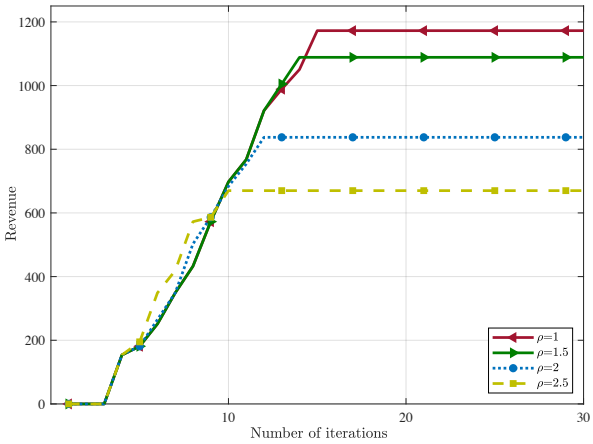


Fig. 7: Revenue versus the number of iterations under different value of ρ.

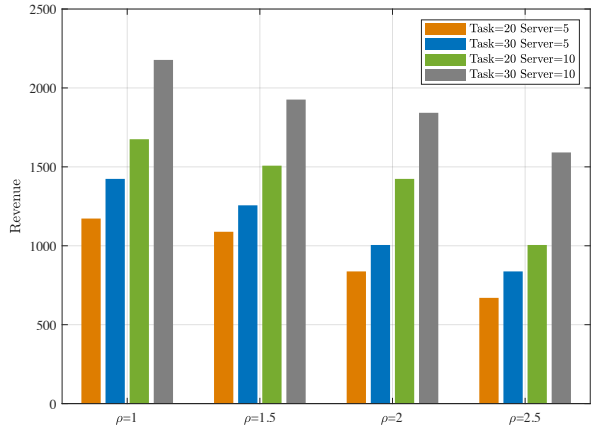


Fig. 8: Revenue versus the value of ρ under different number of tasks and servers.

from Fig.6.

Fig.5 demonstrates the convergence and the revenue of the TORA algorithm versus the number of MEC servers and tasks. It can be seen that the algorithm we proposed owns the fast convergent rate with 20 tasks and 10 servers, because in this case, each server undertakes the least number of tasks. On the contrary, the convergence of the algorithm is the worst when there are 30 tasks and 5 servers, which means each server processes the most tasks on average compared with the other three cases. It is also observed that the TORA algorithm reaches the most revenue with 30 tasks and 10 servers and the minimal revenue with 20 tasks and 5 servers.

Fig.6 shows the performance of the execution time of those four algorithms. In this experiment, we set the number of tasks is 20, located close to 5 MEC servers, so that the execution condition of our designed algorithm can be better measured. It can be seen that the TORA algorithm takes the shortest execution time. The execution time of the partial TORA algorithm and that of the online scheduling algorithm perform nearest. Since the $\alpha-$Fairness algorithm focuses on fairness, it takes the longest execution time. It is also observed that the preemption rule and the resumption rule we proposed do not take extra startup time.

Fig.7 illustrates the convergence of the TORA algorithm versus the value of ρ. In this experiment, we set 5 MEC servers and 20 tasks. It is observed that the algorithm owns the slowest convergent rate and the largest revenue when the value of ρ is equal to 1. On the contrary, the algorithm owns the fastest

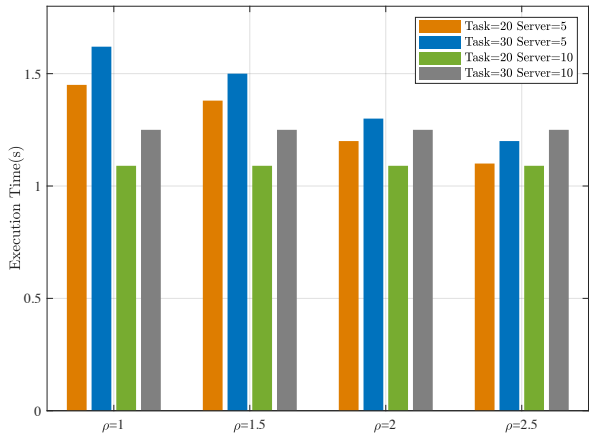


Fig. 9: The execution time versus the value of ρ under different number of tasks and servers.

convergent rate and the minimal revenue when the value of ρ is equal to 2.5. Fig.8 shows the revenue of the TORA algorithm versus the value of ρ under different numbers of tasks and servers. From Fig.8, notice that with the increasing of the value of ρ, regardless of the number of tasks or servers, the revenue will decrease. It is because a larger value of ρ means that the conditions for the resumption of the tasks that are preempted are stricter. Thus, when ρ rises from 1 to 2.5, the number of fully processed tasks is decreasing which leads to lower revenue. Fig.9 demonstrates the performance of the execution time affected by the value of ρ. It can be seen that when the value of ρ increases, the execution time decreases. Similarly, since the increase of the value of ρ causes a decrease in the number of fully processed tasks, the execution time of the TORA algorithm will be shortened.

VII. Conclusion

In this paper, we designed an online data-driven scheduling mechanism to address the computational offloading of deadline-sensitive tasks. Initially, we formulated an online revenue maximization problem to ensure tasks were fully transmitted and computed within their respective deadlines. To tackle this problem, we devised a practical two-step online resource allocation algorithm comprising an online spectrum resource allocation algorithm and an online computing resource allocation algorithm. Additionally, we used duality theory to verify the competitive ratio of the TORA algorithm. Simulations from the revenue increase and execution time improvement demonstrated the superiority of our scheduling mechanism.

Appendix

First, for every task $x_j \in X^p$ and $t \in \Omega_j$, we can get the following geometrical relationship as:

$$\sum_{m\in \mathrm{M}} \int_o^{\infty} \kappa_m(t)dt \le \sum_{m\in \mathrm{M}} (1+\rho) \int_0^{\infty} \kappa_m^{-\rho}(t)dt. \tag{25}$$

Proof: Notice that $\kappa_m(t)$ denotes tasks from task set $x_j \in A(t) \wedge x_j \in X^p$ to be executed at time t, satisfying $\kappa_m(t) \ge \pi_{js}^m(t)/c_{js}^m(t)$ during time interval $t \in \Omega_j$. $\kappa_m^{-\rho}(t)$ denotes tasks from task set $x_j \in A^{-\rho}(t) \wedge x_j \in X^p$ to be executed at time t, satisfying $\kappa_m^{-\rho}(t) \ge \pi_{js}^m(t)/c_{js}^m(t)$ during time interval $t \in \Omega_j^{-\rho}$. Thus, we get that $\kappa_m^{-\rho}(t) \subseteq \kappa_m(t)$, denoting $\kappa_m^{-\rho}(t)$ is the subset of $\kappa_m(t)$ at each time $t \in \Omega_j^{-\rho}$. At the time, if we execute task x_j from task set $\kappa_m(t)$, and task $x_{j'}$ from task set $\kappa_m^{-\rho}(t)$, this inequation $\pi_{j's}^m(t)/c_{j's}^m(t) \le \pi_{js}^m(t)/c_{js}^m(t)$ will be satisfied. Thus, when $t \in \Omega_j$, the following geometrical relationship is derived:

$$\begin{aligned} &\sum_{m\in \mathrm{M}} \frac{a_j - \rho u_j(t) - a_j}{a_j - r_j} \int_o^{\infty} \kappa_m(t)dt \\ &\le \sum_{m\in \mathrm{M}} \int_0^{\infty} \kappa_m^{-\rho}(t)dt. \end{aligned} \tag{26}$$

According to the definition of available time of task, we find that $a_j - r_j \ge (\rho+1)u_j(t)$, getting that $\frac{u_{j(t)}}{a_j - r_j} \le \frac{1}{\rho+1}$. Then by transposition, the formula (26) can be simplified into the followings:

$$\begin{aligned} &\sum_{m\in \mathrm{M}} \int_o^{\infty} \kappa_s(t)dt \\ &\le \frac{1}{(1-\rho\frac{u_{x_j}}{d_{x_j}-a_{x_j}})} \sum_{s\in \mathrm{S}} \int_0^{\infty} \kappa_s^{-\rho}(t)dt. \end{aligned} \tag{27}$$

Thus when the value $\frac{u_{x_j}}{d_{x_j}-a_{x_j}}$ is equal to $\frac{1}{\rho+1}$, the geometrical relationship is satisfied. With the above, this completes the proof of Lemma 1.

References

[1] A. Mukherjee, D. De, N. Dey, R. G. Crespo, and E. Herrera-Viedma, "Disastdrone: A disaster aware consumer internet of drone things system in ultra-low latent 6g network," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 1, pp. 38–48, 2023.

[2] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas, J. S. Thompson, E. G. Larsson, M. D. Renzo, W. Tong, P. Zhu, X. Shen, H. V. Poor, and L. Hanzo, "On the road to 6g: Visions, requirements, key technologies, and testbeds," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

[3] S. He, C. Du, and M. S. Hossain, "6g-enabled consumer electronics device intrusion detection with federated meta-learning and digital twins in a meta-verse environment," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2023.

[4] J. Li, W. Liang, Y. Li, Z. Xu, X. Jia, and S. Guo, "Throughput maximization of delay-aware dnn inference in edge computing by exploring dnn model partitioning and inference parallelism," *IEEE Transactions on Mobile Computing*, vol. 22, no. 5, pp. 3017–3030, 2023.

[5] J. Du, W. Cheng, G. Lu, H. Cao, X. Chu, Z. Zhang, and J. Wang, "Resource pricing and allocation in mec enabled blockchain systems: An a3c deep reinforcement learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 33–44, 2022.

[6] Z. Liu, Q. Li, X. Chen, C. Wu, S. Ishihara, J. Li, and Y. Ji, "Point cloud video streaming: Challenges and solutions," *IEEE Network*, vol. 35, no. 5, pp. 202–209, 2021.

[7] N. Zhao, W. Du, F. Ren, Y. Pei, Y.-C. Liang, and D. Niyato, "Joint task offloading, resource sharing and computation incentive for edge computing networks," *IEEE Communications Letters*, vol. 27, no. 1, pp. 258–262, 2023.

[8] Y. Du, J. Li, L. Shi, T. Liu, F. Shu, and Z. Han, "Two-tier matching game in small cell networks for mobile edge computing," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 254–265, 2022.

[9] J. Du, Z. Kong, A. Sun, J. Kang, D. Niyato, X. Chu, and F. R. Yu, "Maddpg-based joint service placement and task offloading in mec empowered air-ground integrated networks," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

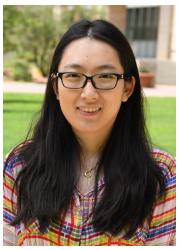
[10] S. Liu, Y. Yu, L. Guo, P. L. Yeoh, B. Vucetic, Y. Li, and T. Q. Duong, "Satisfaction-maximized secure computation offloading in multi-eavesdropper mec networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4227–4241, 2022.
[11] Z. Li, N. Zhu, D. Wu, H. Wang, and R. Wang, "Energy-efficient mobile edge computing under delay constraints," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 2, pp. 776–786, 2022.
[12] Z. Liu, C. Zhan, Y. Cui, C. Wu, and H. Hu, "Robust edge computing in uav systems via scalable computing and cooperative computing," *IEEE Wireless Communications*, vol. 28, no. 5, pp. 36–42, 2021.
[13] X. Shao, G. Hasegawa, M. Dong, Z. Liu, H. Masui, and Y. Ji, "An online orchestration mechanism for general-purpose edge computing," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 927–940, 2023.
[14] P. Dai, K. Hu, X. Wu, H. Xing, F. Teng, and Z. Yu, "A probabilistic approach for cooperative computation offloading in mec-assisted vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 899–911, 2022.
[15] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. Shen, "Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 20, no. 3, pp. 939–951, 2021.
[16] W. Zhang, G. Zhang, and S. Mao, "Joint parallel offloading and load balancing for cooperative-mec systems with delay constraints," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4249–4263, 2022.
[17] H. Maleki, M. Başaran, and L. Durak-Ata, "Handover-enabled dynamic computation offloading for vehicular edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 7, pp. 9394–9405, 2023.
[18] W. Chu, P. Yu, Z. Yu, J. C. Lui, and Y. Lin, "Online optimal service selection, resource allocation and task offloading for multi-access edge computing: A utility-based approach," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 4150–4167, 2023.
[19] Y. Gao, L. Wang, Z. Xie, Z. Qi, and J. Zhou, "Energy- and quality of experience-aware dynamic resource allocation for massively multiplayer online games in heterogeneous cloud computing systems," *IEEE Transactions on Services Computing*, vol. 16, no. 3, pp. 1793–1806, 2023.
[20] G. Cui, Q. He, X. Xia, F. Chen, F. Dong, H. Jin, and Y. Yang, "Ol-eua: Online user allocation for noma-based mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 4, pp. 2295–2306, 2023.
[21] P. Wang, B. Di, L. Song, and N. R. Jennings, "Multi-layer computation offloading in distributed heterogeneous mobile edge computing networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1301–1315, 2022.
[22] H. Tan, S. H.-C. Jiang, Z. Han, and M. Li, "Asymptotically optimal online caching on multiple caches with relaying and bypassing," *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1841–1852, 2021.
[23] J. P. Champati and B. Liang, "Delay and cost optimization in computational offloading systems with unknown task processing times," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1422–1438, 2021.
[24] M. Li and P. Li, "Crowdsourcing in cyber-physical systems: Stochastic optimization with strong stability," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 2, pp. 218–231, 2013.
[25] B. Lucier, I. Menache, J. S. Naor, and J. Yaniv, "Efficient online scheduling for deadline-sensitive jobs: Extended abstract," in *Proceedings of the Twenty-fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, ser. SPAA '13. ACM, 2013, pp. 305–314.
[26] Z. Zheng and N. B. Shroff, "Online multi-resource allocation for deadline sensitive jobs with partial values in the cloud," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.
[27] C. J. Wong, S. Sen, T. Lan, and M. Chiang, "Multiresource allocation: Fairness-efficiency tradeoffs in a unifying framework," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1785–1798, Dec 2013.
[28] C. Liu, K. Wang, H. Zhang, X. Li, and H. Ji, "Rendered tile reuse scheme based on fov prediction for mec-assisted wireless vr service," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 3, pp. 1709–1721, 2023.
[29] Z. Liao, J. Peng, J. Huang, J. Wang, J. Wang, P. K. Sharma, and U. Ghosh, "Distributed probabilistic offloading in edge computing for 6g-enabled massive internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5298–5308, 2021.
[30] Z. Liu, Y. Zhao, J. Song, C. Qiu, X. Chen, and X. Wang, "Learn to coordinate for computation offloading and resource allocation in edge computing: A rational-based distributed approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3136–3151, 2022.

Lichao Yang is an assistant professor in the School of Public Health, Soochow University Medical College. Before that, she received the B.S.degree in Mathematics and Statistics from Zhengzhou University (ZZU), Henan, China, in 2015 and the Ph.D.degree in the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China in 2021. She worked toward the Postdoctoral in the Vanke School of Public Health, Tsinghua University (THU) in 2023. Her current research interests include multiple edge computing, artificial intelligence, public health emergency management, and disease screening and diagnosis.

Kailin Wang received the B.S.degree in communication engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2021. She is currently working toward the Ph.D. degree with the School of Information and Communication Engineering, BUPT, Beijing, China. Her current research interests include integrated sensing and communication, resource management, and future communication networks.

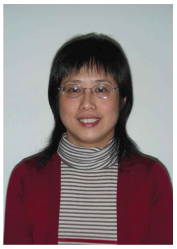
Mingyan Xiao received her Ph.D from University of Texas at Arlington in 2022, M.S. degree from National University of Defense Technology in 2017, and B.E. from Nanjing University of Aeronautics and Astronautics in 2014, respectively. She is currently a tenure track assistant profeesor in the department of Computer Science at California State Polytechnic University, Pomona. Her recent research interests are in the area of data-driven security and privacy.

Heli Zhang received the B.S. degree in communication engineering from Central South University in 2009, and the PH.D. Degree in communication and information System from Beijing University of Posts and Telecommunications (BUPT) in 2014. From 2014 to 2018, she was the Lecturer in the School of Information and Communication Engineering at BUPT. From 2018, she has been the associate professor in the School of Information and Communication Engineering at BUPT. She has been the reviewer for Journals of IEEE Wireless Communications, IEEE Communication Magazine, IEEE Transactions on Vehicular Technology, IEEE Communication Letters and IEEE Transactions on Networking. She participated in many National projects funded by National Science and Technology Major Project, and National Natural Science Foundation of China, and cooperated with many Corporations in research. Her research interests include 5G/6G, mobile edge computing and network security.



Ming Li is an Associate Professor with the Computer Science and Engineering Department at the University of Texas at Arlington. Before that, she worked as an Assistant Professor at the University of Nevada, Reno, for four years (2014 to 2018). She received her Ph.D. at Mississippi State University in 2014, her master's degree from Beijing University of Posts and Telecommunications in 2010, and her bachelor's degree from Sun Yat-Sen University in 2007. Her research interests are in the general areas of networking, mobile systems (e.g., VR), and cybersecurity. She attempts to explore the research topics with approaches rooted in applied AI, sensing, and computing. Her research has been funded by NSF, DoT, and NIH. She is a recipient of the NSF CRII Award (2016), NSF CAREER Award (2020), and N2Women Rising Star in Networking and Communications (2023).

Xi Li received her BE degree and PhD degree from Beijing University of Posts and Telecommunications (BUPT) in the major of communication and information system in 2005 and 2010 respectively. In 2017 and 2018, she was a Visiting Scholar with The University of British Columbia, Vancouver, BC, Canada. She is currently a Professor with the School of Information and Communication Engineering of BUPT. She has published more than 100 papers in international journals and conferences. Her current research interests include resource management and intelligent networking in next generation networks, the Internet of Things, and cloud computing. She has also served as a TPC Member of IEEE WCNC 2012/2014/2015/2016/2019/2020/2021, PIMRC 2017/2018/2019/2020, GLOBECOM 2015/2017/2018, ICC 2015/2016/2017/2018/2019, Infocom 2018 and CloudCom 2013/2014/2015, the Chair of Special Track on cognitive testbed in CHINACOM 2011, the Workshop Chair of IEEE GreenCom 2019, and a Peer Reviewer of many academic journals.



Hong Ji received the B.S. degree in communications engineering and the M.S. and the Ph.D. degrees in information and communications engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1989, 1992, and 2002, respectively. In 2006, she was a Visiting Scholar with The University of British Columbia, Vancouver, BC, Canada. She is currently a Professor with BUPT. She has authored more than 300 journal/conference papers. Several of her papers had been selected for Best paper.