Dynamic Task Offloading and Resource Allocation for NOMA-Aided Mobile Edge Computing: An Energy Efficient Design

Ying Chen, *Senior Member, IEEE*, Jiajie Xu, Yuan Wu, *Senior Member, IEEE*, Jie Gao, *Senior Member, IEEE*, and Lian Zhao, *Fellow, IEEE*

***Abstract*—In recent years, the Internet of Things (IoTs) and mobile communication technologies have developed rapidly. Meanwhile, many delay-sensitive and computation-intensive IoT services have been widely applied. Because of the limited computing resources, storage, and battery capacity of IoT devices, mobile edge computing (MEC) is emerging as a promising paradigm to help process the tasks of IoT devices. Furthermore, non-orthogonal multiple access (NOMA) has evolved as a practical approach to meeting the requirement of massive connectivity. In this article, we study the NOMA-aided dynamic task offloading problem for the IoT, which combines task scheduling and computing resource allocation decisions. We model and formulate the problem as a stochastic optimization problem, and our goal is to minimize the system energy consumption while satisfying performance requirements. We transform the original problem into a deterministic optimization problem through stochastic optimization technology. Then, we decompose it into four sub-problems and propose the energy efficient task offloading (EETO) algorithm to solve these four sub-problems. Our proposed EETO algorithm does not rely on prior statistical knowledge related to task arrival or wireless channel conditions. Through theoretical analysis and experiment results, we demonstrate that our EETO algorithm can make a flexible trade-off between system energy consumption and performance. Additionally, the EETO algorithm can effectively decrease the system energy consumption while ensuring system performance.**

***Index Terms*—Internet of Things (IoT), mobile edge computing (MEC), non -orthogonal multiple access (NOMA), offloading.**

Manuscript received 22 October 2023; revised 5 January 2024; accepted 17 February 2024. Date of publication 12 March 2024; date of current version 8 August 2024. This work was supported in part by the Beijing Natural Science Foundation under Grant L232050, in part by the Project of Cultivation for young top-motch Talents of Beijing Municipal Institutions under Grant BPHR202203225, and the in part by Young Elite Scientists Sponsorship Program by BAST under Grant BYESS2023031, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515011287. *(Corresponding author: Yuan Wu.)*

Ying Chen and Jiajie Xu are with the Computer School, Beijing Information Science and Technology University, Beijing 100101, China (e-mail: chenying@bistu.edu.cn; xujiajie@bistu.edu.cn).

Yuan Wu is with the State Key Lab of Internet of Things for Smart City, University of Macau, Macao 999078, China, and also with the Department of Computer and Information Science, University of Macau, Macao 999078, China (e-mail: yuanwu@um.edu.mo).

Jie Gao is with the School of Information Technology, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: jie.gao6@carleton.ca).

Lian Zhao is with the Department of Electrical and Computer Engineering, Toronto Metropolitan University, Toronto, ON M5B 2K3, Canada (e-mail: l5zhao@torontomu.ca).

Digital Object Identifier 10.1109/TSC.2024.3376240

I. Introduction

With the development and maturity of information technology, the Internet of Things (IoT) devices have been widely deployed [1]. The number of IoT devices around the world will exceed 75 billion by 2025 and will keep growing rapidly in the next decades [2]. Furthermore, applications such as autonomous driving, e-healthcare, augmented reality (AR), and face recognition (FR) have been widely applied. However, the services of these applications are usually delay-sensitive and computation-intensive. It is difficult or even impossible to process all the tasks generated by such applications at IoT devices locally because of the limited battery, storage, and computation capacities [3].

Mobile cloud computing (MCC) is a considerable approach to solving the problems mentioned above [4]. Through offloading to cloud servers that can meet the requirements of computation resources, computation tasks generated at IoT devices can be processed. However, there are several challenges while offloading computation tasks to the cloud servers. Specifically, because of the long distances between remote clouds and IoT devices, it is not easy to satisfy the demands of delay. Besides, offloading a large amount of data to clouds will increase data traffic and can cause a heavy load on the network [5]. Mobile edge computing (MEC) is expected to be an effective solution [6]. MEC allows IoT devices to transmit computation tasks to edge servers (ESs), which are usually placed in small-cell base stations (SBSs) near the IoT devices. Therefore, the challenges to MCC regarding the delay and network load can be effectively resolved by MEC.

With the fast growth in the number of IoT devices and the corresponding data traffic, non-orthogonal multiple access (NOMA) as a prospective method for supporting massive connectivity with limited spectrum resources has attracted enormous attention [7]. Different from traditional orthogonal multiple access, an SBS can serve multiple IoT devices via NOMA simultaneously at the same frequency [8]. This feature of NOMA significantly improves spectral efficiency and satisfies the requirements of high-throughput services. Specifically, NOMA adopts superposition coding at IoT devices and successive interference cancellation (SIC) technologies at SBSs to ensure that multiple IoT devices can communicate with SBS by the same frequency at the same time. In this way, most of the interference

can be eliminated, and the transmission efficiency between IoT devices and SBSs can be greatly improved.

This work studies the energy efficient dynamic task offloading for NOMA-aided MEC system. We jointly optimize the strategies of task scheduling and resource allocation. We optimize the system's energy consumption by incorporating the local side and edge side under the premise of performance guarantee. Since the quality of the wireless channel state and the generation process of computation tasks are dynamic and difficult to predict accurately, the task offloading problem is defined as a stochastic optimization problem. Furthermore, we design and propose a NOMA-aided energy efficient task offloading (EETO) algorithm with the help of stochastic optimization techniques. Finally, to demonstrate the effectiveness of the EETO algorithm, we conduct extensive experiments. The following three points summarize our primary contributions.

- The dynamic task offloading and resource allocation problem for NOMA-aided MEC is studied, and the system framework includes multiple SBSs and a collection of IoT devices. For each SBS, an edge server is equipped to serve the group of IoT devices associated with it, and the IoT devices communicate with the SBS in NOMA manner. The optimization objective is to minimize the system's energy consumption, which includes the local side and ES side, subject to the performance guarantee. We propose a systematic framework to minimize system energy consumption for NOMA-aided MEC. A local computation queue and a local offloading queue are maintained by each IoT device locally, and the tasks created by the IoT device are to be processed by the local CPU or offloaded to the ES. The energy optimization decision variables include: 1) where to place the tasks generated by the IoT device, in the local computation queue or the local offloading queue, 2) the local CPU cycle frequency for processing the computation tasks, 3) the number of computation tasks to be offloaded, 4) the allocation computation resource of edge serves.
- We propose a stochastic optimization-based approach to solve the NOMA-aided task scheduling and resource allocation problem. We first adopt stochastic optimization techniques to reformulate the NOMA-aided energy efficient optimization problem and convert this problem to a deterministic optimization problem. After that, we decompose the deterministic optimization problem into four sub-problems. In addition, we design the EETO algorithm for solving these four sub-problems. Our EETO algorithm can make task scheduling and resource allocation strategies efficiently to dynamically adjust the decision variables for minimizing the system energy consumption without relying on any prior statistical information of the system. Furthermore, the effectiveness of the EETO algorithm is analyzed through strict and meticulous derivations.
- We conduct a series of experiments to validate the performance of the EETO algorithm. In the experiments, our proposed EETO algorithm is compared with both benchmark algorithms and algorithms from related literature. The experimental results verify that our EETO algorithm can decrease the system's energy consumption effectively and ensure the performance of the system.

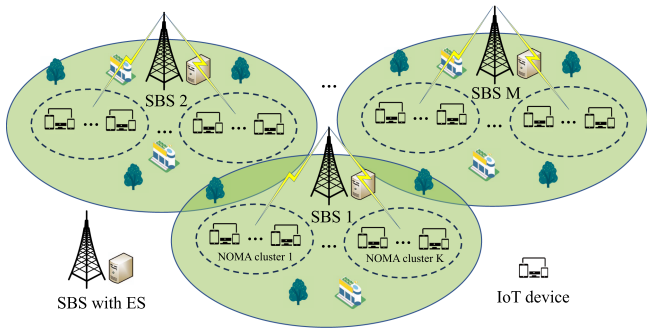


Fig. 1. System model of NOMA-aided MEC.

The rest of this paper is structured as follows. The system model of energy efficient task offloading in NOMA-aided MEC for IoT and the problem formulation are given in Section II. In Section III, we reformulate the problem of task offloading, design the EETO algorithm for addressing the problem, and analyze the performance of the EETO algorithm. Sections IV and V verify the effectiveness of the EETO algorithm and summarize related works, respectively. In Section VI, this paper is summarized and our future work is discussed.

II. System Model and Problem Formulation

A. NOMA-Aided Network Model

As shown in Fig. 1, the dynamic task offloading and resource allocation problem for IoT in the NOMA-aided MEC system is investigated in this paper. There are M SBSs and N IoT devices, denoted collectively by the set $\mathcal{M} = \{1, 2, \ldots, M\}$ and $\mathcal{N} = \{1, 2, \ldots, N\}$. For each SBS m, it deploys an ES to serve the group of associated IoT devices, expressed by $\mathcal{N}^m = \{1, 2, \ldots, N^m\}$. Besides, there are K orthogonal subchannels for each SBS denoted by $\mathcal{K} = \{1, 2, \ldots, K\}$. Let $\mathcal{N}^{m,k} = \{1, 2, \ldots, N^{m,k}\}$ represent the set of IoT devices connected to SBS m and transmitted through subchannel k. With the help of NOMA, the set of IoT devices $\mathcal{N}^{m,k}$ composing NOMA cluster k can use the same spectrum resources to communicate with SBS m through subchannel k simultaneously. We adopt a discrete time slot, the collection of which is denoted as $\mathcal{T} = \{1, 2, \ldots T\}$. The main symbols and their descriptions are enumerated in Table I.

B. Task and Communication Model

Computation tasks are generated by IoT devices at the beginning of each time slot. Let $A_n^{m,k}(t)$ represent the data size of the arrived computation task (in bits) at IoT device n ($\forall n \in \mathcal{N}^{m,k}, k \in \mathcal{K}, m \in \mathcal{M}$). Similar to [9], the binary offloading strategy for the computation tasks is adopted in this paper, that is, the generated tasks are either processed locally or offloaded to SBS. Let $x_n^{m,k}(t) \in \{0, 1\}$ denote the offloading decision of the computation task for IoT device n, where $x_n^{m,k}(t) = 0$ if the computation task is to be processed locally and $x_n^{m,k}(t) = 1$

TABLE I
NOTATIONS

Symbol	Description
$\mathcal{M}$	The set of SBSs
$\mathcal{K}$	The set of subchannels
$\mathcal{N}$	The set of IoT devices
$\mathcal{N}^m$	The collection of IoT devices associated with SBS m
$\mathcal{N}^{m,k}$	The collection of IoT devices associated with SBS m and transmitted through subchannel k
$p_n^{m,k}$	Transmission power
$h_n^{m,k}$	Channel power gain
$f_n^{m,k}$	CPU cycle frequency
$R_n^{m,k}$	Transmission rate
ϕ	CPU cycles needed to process one bit of tasks
ξ	Computation energy efficiency coefficient of IoT device
ϵ	Unit energy consumption of the ES
B	The system uplink Channel bandwidth of SBS
F^m	The computing resources capacities of ES m
$A_n^{m,k}$	Size of the generated computation tasks
$Q_{n,l}^{m,k}$	Length of local computation queue
$Q_{n,o}^{m,k}$	Length of local offloading queue
$Q_{n,b}^{m,k}$	Queue length in ES
$D_{n,l}^{m,k}$	Size of local computation tasks
$D_{n,o}^{m,k}$	Size of offloaded tasks
$D_{n,b}^{m,k}$	Size of tasks processed by ES

if the computation task is to be transmitted to SBS m through subchannel k for processing.

IoT devices communicate with SBS through NOMA manner. Through the SIC technique, received signals from the collection of IoT devices in the NOMA cluster are iteratively decoded by successively canceling the interference. In this paper, we adopt the non-increasing channel gains for decoding the signals of IoT devices $\mathcal{N}^{m,k}$ connected with SBS m transmitted through subchannel k, which is denoted by

$$h_1^{m,k}(t) \geq h_2^{m,k}(t) \geq \ldots \geq h_n^{m,k}(t) \geq \ldots \geq h_{N^{m,k}}^{m,k}(t), \quad \forall m \in \mathcal{M}, \forall k \in \mathcal{K}, \tag{1}$$

where $h_n^{m,k}(t)$ refers to the channel gain of uplink from IoT device n to SBS m through subchannel k. Thus, the corresponding ratio of signal-to-noise is formulated as

$$\gamma_n^{m,k}(t) = \frac{||h_n^{m,k}(t)||^2 p_n^{m,k}(t)}{\sigma^2 + \delta_{n,intra}^{m,k}(t) + \delta_{n,inter}^{m,k}(t)}, \tag{2}$$

where the transmission power is expressed by $p_n^{m,k}(t)$, σ^2 stands for the Gaussian white noise, $\delta_{n,intra}^{m,k}(t) = \sum_{i=n+1}^{N^{m,k}} ||h_i^{m,k}(t)||^2 p_i^{m,k}(t)$ denotes the intra-cell interference and $\delta_{n,inter}^{m,k}(t) = \sum_{s=1,s\neq m}^{M} \sum_{j=1}^{N^{s,k}} ||h_j^{s,k}(t)||^2 p_j^{s,k}(t)$ represents inter-cell interference, respectively. Let B represent the uplink system bandwidth that SBS. Thus, the transmission rate $R_n^{m,k}(t)$ is formulated as

$$R_n^{m,k}(t) = \frac{B}{K} \log(1 + \gamma_n^{m,k}(t)). \tag{3}$$

C. Queuing Model

For each IoT device n ($n \in \mathcal{N}^{m,k}, k \in \mathcal{K}, m \in \mathcal{M}$), we set up two queues locally for each IoT device (i.e., the local computation queue $Q_{n,l}^{m,k}$ and the local offloading queue $Q_{n,o}^{m,k}$ [10], [11]. $Q_{n,l}^{m,k}$ and $Q_{n,o}^{m,k}$ (in bits) are maintained locally, where $Q_{n,l}^{m,k}$ stores the tasks for the local computation and $Q_{n,o}^{m,k}$ stores the tasks for offloading. The tasks in $Q_{n,o}^{m,k}$ will be offloaded to SBS m through wireless channel k for processing. Thus, each SBS m establishes a queue $Q_{n,b}^{m,k}$ (in bits) for IoT device n associated with it.

The size of computation tasks at time slot t that will be calculated locally is represented by $D_{n,l}^{m,k}(t)$. Let ϕ stand for the needed number of CPU cycles to execute each bit of computation task. The specific calculation of $D_{n,l}^{m,k}(t)$ is expressed by

$$D_{n,l}^{m,k}(t) = \frac{\tau f_n^{m,k}(t)}{\phi}, \tag{4}$$

where τ represents the duration of each discrete time slot and $f_n^{m,k}(t)$ is the CPU cycle frequency of IoT device which cannot be greater than the maximum value $f^{\max}$. Therefore, the evolution of $Q_{n,l}^{m,k}$ is expressed by

$$Q_{n,l}^{m,k}(t+1) = (1 - x_n^{m,k}(t)) A_n^{m,k}(t) + \max\{Q_{n,l}^{m,k}(t) - D_{n,l}^{m,k}(t), 0\}. \tag{5}$$

Note that $D_{n,l}^{m,k}(t)$ cannot exceed $Q_{n,l}^{m,k}(t)$, that is, $D_{n,l}^{m,k}(t) \leq Q_{n,l}^{m,k}(t)$.

Regarding the local offloading queue $Q_{n,o}^{m,k}$, let $D_{n,o}^{m,k}(t)$ refer to the number of tasks that are offloaded from IoT device n to SBS m through subchannel k in time slot t. The relation between $D_{n,o}^{m,k}(t)$ and the transmission rate $R_n^{m,k}(t)$ is given by the below constraint

$$D_{n,o}^{m,k}(t) \leq R_n^{m,k}(t)\tau, \quad \forall m \in \mathcal{M}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N}^{m,k}. \tag{6}$$

The evolution of local offloading queue $Q_{n,o}^{m,k}$ is

$$Q_{n,o}^{m,k}(t+1) = x_n^{m,k}(t) A_n^{m,k}(t) + \max\{Q_{n,o}^{m,k}(t) - D_{n,o}^{m,k}(t), 0\}. \tag{7}$$

Note that $D_{n,o}^{m,k}(t)$ cannot exceed $Q_{n,o}^{m,k}(t)$, that is, $D_{n,o}^{m,k}(t) \leq Q_{n,o}^{m,k}(t)$.

Further, similar to [12], [13], we consider that the ES maintains a queue for each IoT device associated with it. Let $D_{n,b}^{m,k}$ represent the number of processed tasks in time slot t by the ES of SBS m in NOMA cluster k for IoT device n, and the

following constraint is required to be satisfied, which is

$$\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} D_{n,b}^{m,k}(t) \le \frac{F^m \tau}{\phi}, \quad \forall m \in \mathcal{M}, \tag{8}$$

where F^m represents the computing resource capacities of ES m.

We define $Q_{n,b}^{m,k}$ as the queue length of IoT device $n \in \mathcal{N}^{m,k}$ at the ES of SBS m. Therefore, the update of $Q_{n,b}^{m,k}$ is given by

$$Q_{n,b}^{m,k}(t+1) = D_{n,o}^{m,k}(t) + \max\{Q_{n,b}^{m,k}(t) - D_{n,b}^{m,k}(t), 0\}. \tag{9}$$

Note that $D_{n,b}^{m,k}(t)$ cannot exceed $Q_{n,b}^{m,k}(t)$, that is, $D_{n,b}^{m,k}(t) \le Q_{n,b}^{m,k}(t)$.

D. Energy Consumption Model

We investigate the system energy consumption including the local computation, transmission, and MEC components.

Let $E_{n,l}^{m,k}(t)$ represent the energy consumption for local execution at IoT device $n \in \mathcal{N}^{m,k}$. Thus, $E_{n,l}^{m,k}(t)$ is given by

$$E_{n,l}^{m,k}(t) = \xi {f_{n,l}^{m,k}(t)}^3 \tau, \tag{10}$$

where ξ represents the coefficient of energy that relies on the architecture of the chip. The energy consumption for transmission, represented by $E_{n,o}^{m,k}(t)$, is formulated by

$$E_{n,o}^{m,k}(t) = \frac{D_{n,o}^{m,k}(t)}{R_n^{m,k}(t)} p_n^{m,k}(t). \tag{11}$$

Let $E_{n,b}^{m,k}(t)$ stand for the energy consumption for calculating tasks from IoT device $n \in \mathcal{N}^{m,k}$ at ES m. Therefore, in time slot t, $E_{n,b}^{m,k}(t)$ is formulated by

$$E_{n,b}^{m,k}(t) = \epsilon D_{n,b}^{m,k} \phi, \tag{12}$$

where ϵ represents the unit energy consumption of the ES.

Accordingly, the overall system energy consumption in time slot t is expressed by

$$E(t) = \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} [E_{n,l}^{m,k}(t) + E_{n,o}^{m,k}(t) + E_{n,b}^{m,k}(t)]. \tag{13}$$

E. Problem Formulation

In this paper, maintaining the stability of queues while minimizing the system's long-term average energy consumption is our optimization goal in this paper. A decision is made for each arriving task on whether it needs to be processed at the corresponding IoT device locally or transmitted to the ES for processing. For $Q_{n,l}^{m,k}$ at each IoT device, the CPU cycle frequency needs to be decided for calculation. As for $Q_{n,o}^{m,k}$, the decision is the number of computation tasks to be offloaded. Regarding the queues maintained by each ES, the ES needs to assign the computing resources for the associated IoT devices. As a result, optimization variables can be denoted by the set $\Gamma(t) = \{\boldsymbol{x}(t), \boldsymbol{f}(t), \boldsymbol{D}_o(t), \boldsymbol{D}_b(t)\}$. The investigated problem can be expressed by

$$\mathcal{P}1: \quad \min_{\Gamma(t)} E = \lim_{T\to\infty} \frac{\sum_{t=0}^{T-1} \mathbb{E}\{E(t)\}}{T}, \tag{14}$$

$$\begin{aligned}
\text{s.t.} \quad & C1: x_n^{m,k}(t) \in \{0,1\}, \\
& C2: 0 \le f_n^{m,k}(t) \le f^{\max}, \\
& C3: 0 \le D_{n,l}^{m,k}(t) \le R_n^{m,k}(t)\tau, \\
& C4: \sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} D_{n,b}^{m,k}(t) \le \frac{F^m\tau}{\phi}, \\
& C5: 0 \le D_{n,l}^{m,k}(t) \le Q_{n,l}^{m,k}(t), \\
& C6: 0 \le D_{n,o}^{m,k}(t) \le Q_{n,o}^{m,k}(t), \\
& C7: 0 \le D_{n,b}^{m,k}(t) \le Q_{n,b}^{m,k}(t), \\
& C8: \lim_{t\to\infty} \frac{\mathbb{E}\{Q_{n,l}^{m,k}(t)\}}{t} = 0, \\
& C9: \lim_{t\to\infty} \frac{\mathbb{E}\{Q_{n,l}^{m,k}(t)\}}{t} = 0, \\
& C10: \lim_{t\to\infty} \frac{\mathbb{E}\{Q_{n,l}^{m,k}(t)\}}{t} = 0, \\
& C11: \forall m \in \mathcal{M}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N}^{m,k}.
\end{aligned}$$

In the above problem, $C1$ is the constraint on the binary offloading decisions for all computation tasks, $C2$ specifies the CPU cycle frequency range, $C3$ limits the number of bits transmitted from each IoT device based on the data transmission rate, $C4$ limits the overall computation resources that can be allocated to the associated IoT devices for each SBS, $C5 \sim C7$ are the constraints of queue length, Constraints $C8 \sim C10$ guarantee queue stability.

Because of the stochastic computation task generation process and the dynamically varying wireless channels, $\mathcal{P}1$ is a typical stochastic optimization problem. Next, we adopt a stochastic optimization framework for solving problem $\mathcal{P}1$.

III. NOMA-Aided Energy Efficient Task Offloading Algorithm Design for Internet of Things

A. Energy Efficient Task Offloading Problem Reformulation for Internet of Things

We adopt stochastic optimization theory for solving the task offloading problem $\mathcal{P}1$. Let $\Phi(t) = \{\boldsymbol{Q}_l(t), \boldsymbol{Q}_o(t), \boldsymbol{Q}_b(t)\}$ represent the vector of the current queue backlog, which is composed of the backlog in $Q_{n,l}^{m,k}(t)$ at IoT device n, $Q_{n,o}^{m,k}(t)$ at IoT device n, $Q_{n,b}^{m,k}(t)$ at SBS m, respectively. Therefore, we introduce the quadratic Lyapunov function to illustrate the queue

congestion state of the system. The function is expressed by

$$\Lambda(\Phi(t)) = 0.5 \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{n=1}^{N^{m,k}} [Q_{n,l}^{m,k}(t)^2 + Q_{n,o}^{m,k}(t)^2 + Q_{n,b}^{m,k}(t)^2], \quad (15)$$

which represents the overall backlog of the queues in the system. The function of Lyapunov drift is defined as

$$\Delta(\Phi(t)) = \mathbb{E}\{\Lambda(\Phi(t+1)) - \Lambda(\Phi(t))|\Phi(t)\}. \quad (16)$$

For balancing the queue backlog and energy consumption, the function of drift-plus-penalty is formulated by

$$\Delta_V(\Phi(t)) = V\mathbb{E}\{E(t)|\Phi(t)\} + \Delta(\Phi(t)), \quad (17)$$

where V is a positive tradeoff parameter for weighing the queue backlog and energy consumption.

In the deterministic optimization problem corresponding to $\mathcal{P}1$, our goal is to minimize $\Delta_V(\Phi(t))$ for achieving the optimal energy consumption and meanwhile the backlog of queues at a stable level for each time slot. Instead of directly minimizing (17), we derive a theoretical upper bound of (17) in Theorem 1.

Theorem 1: Upper bound of (17) is

$$\begin{aligned} &V\mathbb{E}\{E(t)|\Phi(t)\} + \Delta(\Phi(t)) \le \aleph + V\mathbb{E}\{E(t)|\Phi(t)\} \\ &+ \mathbb{E}\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{k,m}} Q_{n,l}^{m,k}(t)[(1-x_n^{m,k}(t))A_n^{m,k}(t) \\ &- D_{n,l}^{m,k}(t)]|\Phi(t)\} \\ &+ \mathbb{E}\left\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} Q_{n,o}^{m,k}(t)[x_n^{m,k}(t)A_n^{m,k}(t) \right. \\ &\left. -D_{n,o}^{m,k}(t)]|\Phi(t)\right\} \\ &+ \mathbb{E}\left\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} Q_{n,b}^{m,k}(t)[D_{n,o}^{m,k}(t) - D_{n,b}^{m,k}(t)]|\Phi(t)\right\}, \end{aligned} \quad (18)$$

where $\aleph = 0.5\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}}[(A_n^{m,k,\max})^2 + (D_{n,l}^{m,k,\max})^2 + 2(D_{n,o}^{m,k,\max})^2 + (D_{n,b}^{m,k,\max})^2] \ge 0.5\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}}[A_n^{m,k}(t)^2 + D_{n,l}^{m,k}(t)^2 + 2D_{n,o}^{m,k}(t)^2 + D_{n,b}^{m,k}(t)^2]$, $A_n^{m,k\max}$, $D_{n,l}^{m,k,\max}$, $D_{n,o}^{m,k,\max}$ and $D_{n,b}^{m,k,\max}$ are the upper bounds of $A_n^{m,k}$, $D_{n,l}^{m,k}$, $D_{n,o}^{m,k}$ and $D_{n,b}^{m,k}$, respectively.

Proof: On the basis of (5), (7), (9) and $([x-y]^+ + z)^2 \le x^2 + y^2 + z^2 + 2x(z-y)$, we can achieve

$$\begin{aligned} Q_{n,l}^{m,k}(t+1)^2 &\le Q_{n,l}^{m,k}(t)^2 + [(1-x_n^{m,k}(t))A_n^{m,k}(t)]^2 \\ &+ D_{n,l}^{m,k}(t)^2 + 2Q_{n,l}^{m,k}(t)[(1-x_n^{m,k}(t)) \\ &\times A_n^{m,k}(t) - D_{n,l}^{m,k}(t)], \end{aligned} \quad (19)$$

$$\begin{aligned} Q_{n,o}^{m,k}(t+1)^2 &\le Q_{n,o}^{m,k}(t)^2 + [x_n^{m,k}(t)A_n^{m,k}(t)]^2 + D_{n,o}^{m,k}(t)^2 \\ &+ 2Q_{n,o}^{m,k}(t)[x_n^{m,k}(t)A_n^{m,k}(t) - D_{n,o}^{m,k}(t)], \end{aligned} \quad (20)$$

$$\begin{aligned} Q_{n,b}^{m,k}(t+1)^2 &\le Q_{n,b}^{m,k}(t)^2 + D_{n,o}^{m,k}(t)^2 + D_{n,b}^{m,k}(t)^2 \\ &+ 2Q_{n,b}^{m,k}(t)[D_{n,o}^{m,k}(t) - D_{n,b}^{m,k}(t)]. \end{aligned} \quad (21)$$

Combining (19), (20), and (21), we can obtain

$$\begin{aligned} &V\mathbb{E}\{E(t)|\Phi(t)\} + \Delta(\Phi(t)) \le \aleph + V\mathbb{E}\{E(t)|\Phi(t)\} \\ &+ \mathbb{E}\left\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} Q_{n,l}^{m,k}(t)[(1-x_n^{m,k}(t))A_n^{m,k}(t) \right. \\ &\left. -D_{n,l}^{m,k}(t)]|\Phi(t)\right\} \\ &+ \mathbb{E}\left\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} Q_{n,o}^{m,k}(t)[x_n^{m,k}(t)A_n^{m,k}(t) \right. \\ &\left. - D_{n,o}^{m,k}(t)]|\Phi(t)\right\} \\ &+ \mathbb{E}\left\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} Q_{n,b}^{m,k}(t)[D_{n,o}^{m,k}(t) - D_{n,b}^{m,k}(t)]|\Phi(t)\right\}. \end{aligned} \quad (22)$$

□

B. NOMA-Aided Energy Efficient Task Offloading Algorithm

We present a NOMA-aided energy efficient task offloading algorithm for minimizing the upper bound of (18). Because $\aleph$ is a constant, we can transform $\mathcal{P}1$ into a collection of deterministic problems, one for each time slot. The deterministic problem for time slot t, referred to as $\mathcal{P}2$, is expressed as

$$\begin{aligned} \mathcal{P}2: \quad &\min_{\Gamma(t)} \left\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} [Q_{n,l}^{m,k}(t)(1-x_n^{m,k}(t))A_n^{m,k}(t) \right. \\ &+ Q_{n,o}^{m,k}(t)x_n^{m,k}(t)A_n^{m,k}(t)] \\ &+ \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} \left[V\xi f_n^{m,k}(t)^3\tau - Q_{n,l}^{m,k}(t)\frac{f_n^{m,k}(t)\tau}{\phi}\right] \\ &+ \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} \left[Vp_n^{m,k}(t)\frac{D_{n,o}^{m,k}(t)}{R_n^{m,k}(t)} - Q_{n,o}^{m,k}(t)D_{n,o}^{m,k}(t) \right. \\ &\left. +Q_{n,b}^{m,k}(t)D_{n,o}^{m,k}(t)\right] \\ &\left. + \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} [V\epsilon D_{n,b}^{m,k}(t)\phi - Q_{n,b}^{m,k}(t)D_{n,b}^{m,k}(t)]\right\}, \end{aligned} \quad (23)$$

$$\begin{aligned} \text{s.t.} \quad &C1: x_n^{m,k}(t) \in \{0,1\}, \\ &C2: 0 \le f_n^{m,k}(t) \le f^{\max}, \\ &C3: 0 \le D_{n,l}^{m,k}(t) \le R_n^{m,k}(t)\tau, \end{aligned}$$

$$C4: \sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} D_{n,b}^{m,k}(t) \le \frac{F^m\tau}{\phi},$$
$$C5: 0 \le D_{n,l}^{m,k}(t) \le Q_{n,l}^{m,k}(t),$$
$$C6: 0 \le D_{n,o}^{m,k}(t) \le Q_{n,o}^{m,k}(t),$$
$$C7: 0 \le D_{n,b}^{m,k}(t) \le Q_{n,b}^{m,k}(t),$$
$$C8: \forall m \in \mathcal{M}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N}^{m,k}.$$

We can notice that $\boldsymbol{x}(t)$, $\boldsymbol{f}(t)$, $\boldsymbol{D}_o(t)$ and $\boldsymbol{D}_b(t)$ in problem $\mathcal{P}2$ are decoupled. Hence, $\mathcal{P}2$ can be decomposed into four sub-problems. Next, we describe these sub-problems and their solutions one by one.

1) Offloading Decision: By separating offloading decision $\boldsymbol{x}(t)$ from $\mathcal{P}2$, we can obtain the objective function $\mathcal{P}_{2-1}$:

$$\mathcal{P}_{2-1}: \min_{\boldsymbol{x}(t)} \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} [Q_{n,l}^{m,k}(t)(1-x_n^{m,k}(t))A_n^{m,k}(t) + Q_{n,o}^{m,k}(t)x_n^{m,k}(t)A_n^{m,k}(t)],$$
$$\text{s.t.} \quad x_n^{m,k}(t) \in \{0,1\},$$
$$\forall m \in \mathcal{M}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N}^{m,k}. \tag{24}$$

For each IoT device, the offloading decision is decoupled and the sub-problem is a problem of zero-one integer programming. Hence, the optimal solution $x_n^{m,k}(t)^*$ is expressed by

$$x_n^{m,k}(t)^* = \begin{cases} 0, & \text{if } \ Q_{n,o}^{m,k}(t) \ge Q_{n,l}^{m,k}(t), \\ 1, & \text{otherwise.} \end{cases} \tag{25}$$

2) Local Computation Resource Allocation: $\mathcal{P}_{2-2}$ represents the sub-problem for the allocation of local CPU cycle frequency, which can be formulated by

$$\mathcal{P}_{2-2}: \min_{\boldsymbol{f}(t)} \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} [V\xi f_n^{m,k}(t)^3\tau - Q_{n,l}^{m,k}(t)\frac{f_n^{m,k}(t)\tau}{\phi}],$$
$$s.t. \quad 0 \le f_n^{m,k}(t) \le f^{\max},$$
$$0 \le D_{n,l}^{m,k}(t) \le Q_{n,l}^{m,k}(t),$$
$$\forall m \in \mathcal{M}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N}^{m,k}. \tag{26}$$

$\mathcal{P}_{2-2}$ is seen as a convex problem. We can obtain the optimal solution of local CPU cycle frequency, which can be expressed as

$$f_n^{m,k}(t)^* = \begin{cases} \sqrt{\frac{Q_{n,l}^{m,k}(t)}{3V\phi\xi}}, & \text{if } \ 0 \le \sqrt{\frac{Q_{n,l}^{m,k}(t)}{3V\phi\xi}} \le f^{\max}, \\ f^{\max}, & \text{otherwise.} \end{cases} \tag{27}$$

3) Offloading Computation Allocation: By extracting the items related to the amount of offloaded tasks, we can get

$$\mathcal{P}_{2-3}: \min_{\boldsymbol{D}_o(t)} \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} \left[Vp_n^{m,k}(t)\frac{D_{n,o}^{m,k}(t)}{R_n^{m,k}(t)} - Q_{n,o}^{m,k}(t)D_{n,o}^{m,k}(t) + Q_{n,b}^{m,k}(t)D_{n,o}^{m,k}(t)\right],$$
$$\text{s.t.} \quad 0 \le D_{n,o}^{m,k}(t) \le R_n^{m,k}(t)\tau,$$
$$0 \le D_{n,o}^{m,k}(t) \le Q_{n,o}^{m,k}(t),$$
$$\forall m \in \mathcal{M}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N}^{m,k}. \tag{28}$$

Similar to $\mathcal{P}_{2-1}$, the decisions for the task offloading size (i.e., how many bits to offload in time slot t) are also decoupled for each IoT device in $\mathcal{P}_{2-3}$. Furthermore, this sub-problem is a linear programming problem. Let $G_n^{m,k}(t) = \frac{Vp_n^{m,k}(t)}{R_n^{m,k}(t)} - Q_{n,o}^{m,k}(t) + Q_{n,b}^{m,k}(t)$, then the optimal offloading size $D_{n,o}^{m,k}(t)^*$ is found as

$$D_{n,o}^{m,k}(t)^* = \begin{cases} \min\{R_n^{m,k}(t)\tau, Q_{n,o}^{m,k}(t)\}, & \text{if } \ G_n^{m,k}(t) \le 0, \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

4) ES Computation Resource Allocation: $\mathcal{P}_{2-4}$ gives the problem of ES computation resource allocation.

$$\mathcal{P}_{2-4}: \min_{\boldsymbol{D}_b(t)} \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} [V\epsilon D_{n,b}^{m,k}(t)\phi - Q_{n,b}^{m,k}(t)D_{n,b}^{m,k}(t)],$$
$$\text{s.t.} \sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} D_{n,b}^{m,k}(t) \le \frac{F^m\tau}{\phi},$$
$$0 \le D_{n,b}^{m,k}(t) \le Q_{n,b}^{m,k}(t),$$
$$\forall m \in \mathcal{M}, \forall k \in \mathcal{K}, \forall n \in \mathcal{N}^{m,k}. \tag{30}$$

Note that the computation resource allocation of different edge servers is decoupled. Let $H_n^{m,k}(t) = V\epsilon\phi - Q_{n,b}^{m,k}(t)$. We can find that the items in $H_n^{m,k}(t)$ are not related to the decision variable $D_{n,b}^{m,k}(t)$. Thus, for each ES m, the sub-problem is expressed by

$$\mathcal{P}'_{2-4}: \min_{\boldsymbol{D}_b^m(t)} \sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} H_n^{m,k}(t)D_{n,b}^{m,k}(t),$$
$$\text{s.t.} \sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} D_{n,b}^{m,k}(t) \le \frac{F^m\tau}{\phi},$$
$$0 \le D_{n,b}^{m,k}(t) \le Q_{n,b}^{m,k}(t),$$
$$\forall k \in \mathcal{K}, \forall n \in \mathcal{N}^{m,k}. \tag{31}$$

It can be found that the sub-problem $\mathcal{P}'_{2-4}$ is a knapsack problem, where $\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} D_{n,b}^{m,k}(t) \le \frac{F^m\tau}{\phi}$ stands for the maximum knapsack capacity is $\frac{F^m\tau}{\phi}$ and $H_n^{m,k}(t)$ (where $k \in \mathcal{K}, n \in \mathcal{N}^{m,k}$) is considered as the unit value for the item. Because our

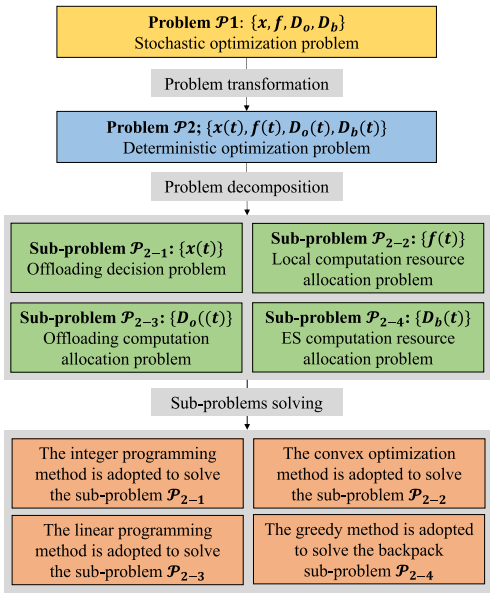


Fig. 2. Relationship among optimization problems.

optimization goal is to minimize sub-problem $\mathcal{P}'_{2-4}$, we need to sort $H_n^{m,k}(t)$ in a non-decreasing order. Then, the solution for the sub-problem is to choose the item with the smallest and negative unit value until there are no non-positive items or the backpack is filled. The following are the detailed steps for solving sub-problem $\mathcal{P}'_{2-4}$, and each ES adopts the same strategy.

1) Define $C^m(t)$ as the available computation resources in ES m and initialize $C^m(t)$ to be $\frac{F^m\tau}{\phi}$.
2) Calculate $H_n^{m,k}(t)$ and sort them in a non-decreasing order.
3) Allocate the computation resources to the group of IoT devices in the set $\mathcal{N}^m$, starting from the device with the lowest $H_n^{m,k}(t)$ (where $k \in \mathcal{K}, n \in \mathcal{N}^{m,k}$). The computation resources that ES of SBS m assigns to IoT device n of NOMA cluster k is

$$D_{n,b}^{m,k}(t)^* = \begin{cases} \min\{Q_{n,b}^{m,k}(t), C^m(t)\}, & \text{if } H_{n^*}^{m,k}(t) \leq 0, \\ 0, & \text{otherwise}, \end{cases} \tag{32}$$

where $n^* = \arg\min H_n^{m,k}(t)$, for $k \in \mathcal{K}, n \in \mathcal{N}^{m,k}$. Remove n^* from set $\mathcal{N}^m$.
4) Update the available computation resources of ES m by $C^m(t) \Leftarrow C^m(t) - D_{n,b}^{m,k}(t)^*$.
5) Loop steps (3) and (4) until the set $\mathcal{N}^m$ is empty.

After solving all four sub-problems, the IoT devices and SBSs execute corresponding offloading and processing according to the obtained solution. The details of the NOMA-aided energy efficient task offloading (EETO) algorithm are given in Algorithm 1. For time slot t, the input of our EETO algorithm is $A_n^{m,k}(t)$, $Q_{n,l}^{m,k}(t)$, $Q_{n,o}^{m,k}(t)$ and $Q_{n,b}^{m,k}(t)$, and the output is the set of decision variables which include $x_n^{m,k}(t)$, $f_n^{m,k}(t)$, $D_{n,o}^{m,k}(t)$ and $D_{n,b}^{m,k}(t)$ (for $m \in \mathcal{M}, k \in \mathcal{K}, n \in \mathcal{N}^{m,k}$). Algorithm 1 contains two loops, for the first loop (lines 1–12), we can obtain the offloading decision, local computation resource allocation, and offloading computation allocation for each IoT device in time slot t. As for the second loop (lines 13–21), we can obtain the ES computation resource allocation of each IoT device.

Algorithm 1: NOMA-aided Energy Efficient Task Offloading (EETO) Algorithm.

Require: $\boldsymbol{A}(t)$, $\boldsymbol{Q}_l(t)$, $\boldsymbol{Q}_o(t)$ and $\boldsymbol{Q}_b(t)$;
Output: $\boldsymbol{x}(t)$, $\boldsymbol{f}(t)$, $\boldsymbol{D}_o(t)$ and $\boldsymbol{D}_b(t)$;
1: **for** $m \in \mathcal{M}$ **do**
2: **for** $k \in \mathcal{K}$ **do**
3: **for** $n \in \mathcal{N}^{m,k}$ **do**
4: **Offloading decision:**
5: Determine the optimal offloading decision $x_n^{m,k}(t)^*$ according to (25);
6: **Local Computation Resource Allocation:**
7: Obtain the optimal local CPU cycle frequency $f_n^{m,k}(t)^*$ in accordance with (27);
8: **Offloading Computation Allocation:**
9: Calculate the optimal computation tasks offloading amount $D_{n,o}^{m,k}(t)^*$ in line with (29).
10: **end for**
11: **end for**
12: **end for**
13: **for** $m \in \mathcal{M}$ **do**
14: Initialize the available computation resources $C^m(t) = \frac{F^m\tau}{\phi}$;
15: Sort $H_n^{m,k}(t)$ for all $n \in \mathcal{N}^{m,k}, k \in \mathcal{K}$ in a non-decreasing order.
16: **for** $n \in \mathcal{N}^m$ **do**
17: **ES Computation Resource Allocation:**
18: Allocate the computation resources ratio $D_{n,b}^{m,k}(t)^*$ to IoT device n according to (32) with the sorted order;
19: Update $C^m(t)$ based on $C^m(t) = C^m(t) - D_{n,b}^{m,k}(t)^*$.
20: **end for**
21: **end for**

Then, we give the time complexity of the EETO algorithm. For the first loop (lines 1–12) in Algorithm 1, the execution time of (25), (27), and (29) is constant. Hence, the time complexity of the first loop is $\mathcal{O}(MKN^{m,k})$. However, each IoT device can make these three decisions independently, so the time complexity of the first loop can be decreased to $\mathcal{O}(1)$. For the second loop (lines 13–21), the time complexity of sorting (line 15) is $O(N^m \log N^m)$ and the allocation of ES computing resource is $\mathcal{O}(N^m)$ which is less than $\mathcal{O}(N^m \log N^m)$. Therefore, the time complexity of the second loop is $\mathcal{O}(MN^m \log N^m)$. Further, because of each ES can make these decisions in parallel, the time complexity can be reduced to $\mathcal{O}(N^m \log N^m)$. According to the above analysis result, the time complexity of our EETO algorithm is $\mathcal{O}(N^m \log N^m)$.

To clearly illustrate the process, as shown in Fig. 2, we present the process of transforming the stochastic optimization problem $\mathcal{P}1$ into static optimization problem $\mathcal{P}2$, which is then decoupled into multiple sub-problems and solved one by one.

C. Optimality Analysis

We conduct a mathematical analysis of our proposed NOMA-aided energy efficient task offloading algorithm in this subsection. Lemma 1 demonstrates the performance of the EETO algorithm as below.

Lemma 1: There is an optimal strategy π^* for different computation task arrival rate η. The strategy meets

$$\begin{aligned}
&\mathbb{E}\{E^{\pi^*}(t)\} = E^*(\eta),\\
&\mathbb{E}\{(1-x_n^{m,k,\pi^*}(t))A_n^{m,k,\pi^*}(t)\} \le \mathbb{E}\{D_{n,l}^{m,k,\pi^*}(t)\},\\
&\mathbb{E}\{x_n^{m,k,\pi^*}(t)A_n^{m,k,\pi^*}(t)\} \le \mathbb{E}\{D_{n,o}^{m,k,\pi^*}(t)\},\\
&\mathbb{E}\{D_{n,o}^{m,k,\pi^*}(t)\} \le \mathbb{E}\{D_{n,b}^{m,k,\pi^*}(t)\},
\end{aligned} \tag{33}$$

where $E^*(\eta)$ denotes the minimum energy consumption with computation task arrival rates η.

Proof: Similar to the [14], Caratheodory's theorem is used for proving Lemma 1. We omit the details of the proof for the sake of brevity. □

The computation task arrival rate has an upper limit for each IoT device. Hence, let $\hat{E}$ and $\check{E}$ represent the value of the upper limit and lower limit for the system energy consumption, respectively. Next, Theorem 2 gives the upper bound of the long-term average energy consumption and queue length on account of Lemma 1.

Theorem 2: Let $\eta+\varrho$ represent the computation task arrival rate, where ϱ is a non-negative value. For a given V, the energy consumption's upper bound is expressed by

$$E^{EETO} \le E^* + \frac{\aleph}{V}. \tag{34}$$

The maximum of the long-term average queue backlog can be written by

$$\bar{Q} \le \frac{\aleph + V(\hat{E}-\check{E})}{\varrho}, \tag{35}$$

where E^* refers to the minimum energy consumed by the system in the case of computation task arrival rate, and $\aleph$ is a constant.

Proof: For a randomized strategy π and computation task arrive rate $\eta+\varrho$, we obtain the following formulas based on Lemma 1.

$$\begin{aligned}
&\mathbb{E}\{E^{\pi}(t)\} = E^*(\eta+\varrho),\\
&\mathbb{E}\{(1-x_n^{m,k,\pi}(t))A_n^{m,k,\pi}(t)\} + \varrho \le \mathbb{E}\{D_{n,l}^{m,k,\pi}(t)\},\\
&\mathbb{E}\{x_n^{m,k,\pi}(t)A_n^{m,k,\pi}(t)\} + \varrho \le \mathbb{E}\{D_{n,o}^{m,k,\pi}(t)\},\\
&\mathbb{E}\{D_{n,o}^{m,k,\pi}(t)\} + \varrho \le \mathbb{E}\{D_{n,b}^{m,k,\pi}(t)\}.
\end{aligned} \tag{36}$$

Considering the strategy π of our EETO algorithm, the goal is to minimize (17). It holds that

$$\begin{aligned}
&\Delta_V(\Phi(t)) \le \aleph + V\mathbb{E}\{E(t)|\Phi(t)\}\\
&+\mathbb{E}\left\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} Q_{n,l}^{m,k}(t)[(1-x_n^{m,k,\pi}(t))A_n^{m,k,\pi}(t)\right.\\
&\left.-D_{n,l}^{m,k,\pi}(t)]\right\}\\
&+\mathbb{E}\left\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} Q_{n,o}^{m,k}(t)[x_n^{m,k,\pi}(t)A_n^{m,k,\pi}(t)\right.\\
&\left.-D_{n,o}^{m,k,\pi}(t)]\right\}\\
&+\mathbb{E}\left\{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}} Q_{n,b}^{m,k}(t)[D_{n,o}^{m,k,\pi}(t)-D_{n,b}^{m,k,\pi}(t)]\right\}
\end{aligned} \tag{37}$$

Plugging (36) into (37), it holds

$$\begin{aligned}
&\mathbb{E}\{\Lambda(\Phi(t+1))-\Lambda(\Phi(t)\} + V\mathbb{E}(E(t)) \le \aleph + VE^*(\eta+\varrho)\\
&\quad-\varrho\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}}\mathbb{E}\{Q_{n,l}^{m,k}(t)+Q_{n,o}^{m,k}(t)+Q_{n,b}^{m,k}(t)\}.
\end{aligned} \tag{38}$$

Taking the expectation on both sides of (38) and summarizing it over all the time slots, we have

$$\begin{aligned}
&V\sum_{t=1}^{T-1}\mathbb{E}(E(t)) \le \aleph T + VTE^*(\eta+\varrho)\\
&\quad-\varrho\sum_{t=1}^{T-1}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}}\mathbb{E}\{Q_{n,l}^{m,k}(t)+Q_{n,o}^{m,k}(t)+Q_{n,b}^{m,k}(t)\}.
\end{aligned} \tag{39}$$

Since $Q_{n,l}^{m,k}(t)$, $Q_{n,o}^{m,k}(t)$, $Q_{n,b}^{m,k}(t)$ and ϱ are positive, the following equation can be achieved

$$V\sum_{t=1}^{T-1}\mathbb{E}(E(t)) \le \aleph T + VTE^*(\eta+\varrho). \tag{40}$$

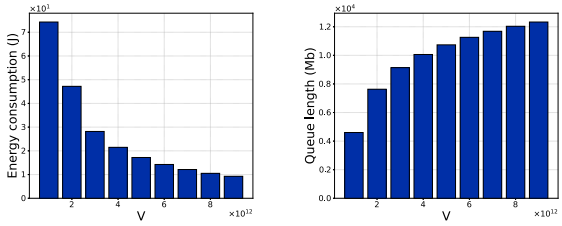
Further, both sides of (40) are divided by VT. When $T\to\infty$ and $\varrho\to 0$, we can get (34).

Through (39), we can also get

$$\begin{aligned}
&\varrho\sum_{t=1}^{T-1}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}}\mathbb{E}\{Q_{n,l}^{m,k}(t)+Q_{n,o}^{m,k}(t)+Q_{n,b}^{m,k}(t)\} \le\\
&\aleph T + VTE^*(\eta+\varrho) - V\sum_{t=1}^{T-1}\mathbb{E}(E(t)).
\end{aligned} \tag{41}$$

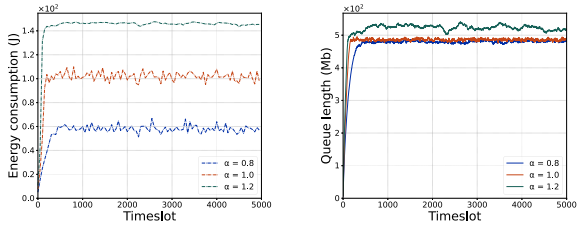
Because $\mathbb{E}(E(t))$ is positive, we can get

$$\begin{aligned}
&\varrho\sum_{t=1}^{T-1}\sum_{k=1}^{K}\sum_{n=1}^{N^{m,k}}\mathbb{E}\{Q_{n,l}^{m,k}(t)+Q_{n,o}^{m,k}(t)+Q_{n,b}^{m,k}(t)\} \le\\
&\aleph T + VT(\hat{E}-\check{E}).
\end{aligned} \tag{42}$$



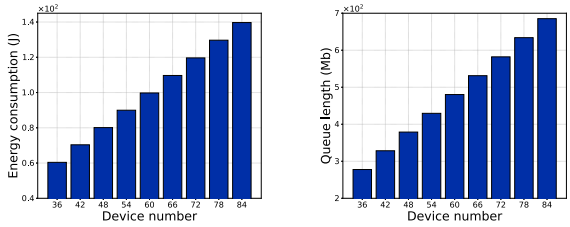
(a) (b)

Fig. 3. Energy consumption and queue length versus control parameter. (a) Energy consumption with different control parameters. (b) Queue length with different control parameters.



(a) (b)

Fig. 4. Energy consumption and queue length versus task arrival rate. (a) Energy consumption with different time slots. (b) Queue length with different time slots.



(a) (b)

Fig. 5. Energy consumption and queue length versus number of IoT devices. (a) Energy consumption with different numbers of devices. (b) Queue length with different numbers of devices.

When $T \to \infty$, we divide (42) by ϱT, and (35) can be obtained. □

IV. Performance Evaluation

A. Experiment Settings

In the experiments, three SBSs are serving multiple IoT devices. The computation task arrival rate $A_n^{m,k}$ is drawn from a uniform distribution over $[0, 2.45 \times 10^5]$ bits. The maximum CPU cycle frequency $f^{\max}$ is 1 GHz [15]. In addition, the needed CPU cycles for processing each bit of computation tasks ϕ is 10^4 cycles, and the energy coefficient ξ is 10^{-27} [16]. The computation capacity of ES is 10 GHz, the bandwidth of each SBS assigned to the set of associated IoT devices is 10 MHz, and each NOMA cluster contains 2 IoT devices [17]. The length of each time slot is 1 s.

B. Parameter Analysis

1) Impact of Parameter V: Fig. 3 illustrates the impact of the tradeoff parameter V on the system energy consumption and long-term average queue length for all IoT devices, respectively. With the rise of V, we can find that the energy consumed by the system decreases over time in Fig. 3(a). The results indicate that a larger V leads the system to emphasize energy consumption, which conforms with (34). Fig. 3(b) illustrates the change of the long-term average queue length with parameter V. It can be found that the total average queue length also rises when V increases, which conforms with (35) in Theorem 2. We can infer that with the increase of V, our presented EETO algorithm tends to weigh the energy consumed by the system more compared with the queue length from these two figures. The tradeoff between energy consumed by the system and queue length means that we can change the value of parameter V to adjust the priority between energy consumption and queue length. A larger V expresses that we care more about the system's energy consumption.

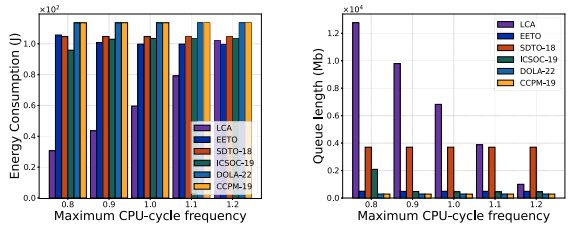
2) Impact of Computation Task Arrival Rate: As shown in Fig. 4, we plot the evolution of total energy consumed by the system and total queue length over time. Let $\alpha \cdot A_n^{m,k}(t)$ represent the computation task arrival rate, where the value of α is 0.8, 1.0, and 1.2, respectively. For each point in Fig. 4(a), the value of the y-axis is averaged. We can notice that the energy consumed by the system rises with the increase in arrival rate, which is because a higher arrival rate will cause more computation tasks to be calculated locally. The reason for the continuous fluctuation is that the task arrival process is uncertain and random, and the arrival of tasks at different time slots may be bursty and fluctuant. The total average queue length in Fig. 4(b) also grows with the rise of the computation task arrival rate. Nevertheless, we can clearly notice that the total average queue length will converge in all these cases. The results verify that our proposed EETO algorithm can coordinate strategy for different computation task arrival rates to guarantee queue stability.

3) Impact of Device Number: We plot the changes in the number of IoT devices on the long-term average energy consumed by the system and queue length in Fig. 5(a) and (b), respectively. The number of devices ranges from 36 to 84 with the step of 6. Fig. 5(a) shows that the size of computation tasks rises when the number of IoT devices increases, which results in an increase in system energy consumption ultimately. Fig. 5(b) depicts that, as the number of IoT devices increases, the queue backlog also rises. For one thing, the increase in the number of IoT devices causes a rise in the number of queues, which further leads to a rise in the length of queues. Besides, because of the limited computation resources, more computation tasks will be waiting in the queues to be calculated.

C. Comparison Experiments

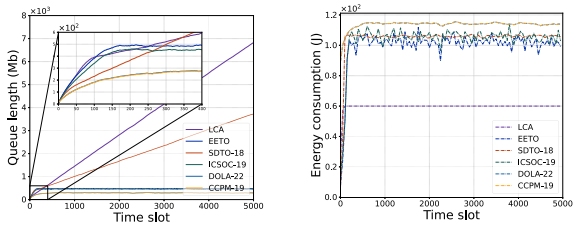
In this subsection, the EETO algorithm is compared with five other algorithms for evaluating the performance in the energy consumption and queue backlog.

- *Local Computation All (LCA):* All the computation tasks created by IoT devices are pushed into the corresponding local computation queue.
- *SDTO-18:* Similar to the stochastic offloading strategy in [18], the arrived computation tasks are randomly pushed into two local queues.



(a) (b)

Fig. 6. Energy consumption and queue length for different algorithms versus maximum CPU-cycle frequencies. (a) Energy consumption with different maximum CPU-cycle frequencies. (b) Queue length with different maximum CPU-cycle frequencies.



(a) (b)

Fig. 7. Energy consumption and queue length for different algorithms versus time slots. (a) Queue length with different time slots. (b) Energy consumption with different time slots.

- *ICSOC-19:* Extended from [19], the offloading decision of each IoT devices using the greedy policy.
- *DOLA-22:* Inspired by [20], the server resource allocation strategy of the DOLA-22 algorithm is allocating the resource based on the proportion of queue backlog.
- *CCPM-19:* The CCPM-19 algorithm is extended from [21]. ES sorts the queue backlog for each IoT device and allocates computing resources with the order.

From Fig. 6(a) and (b), we can find that compared to the SDTO-18 algorithm, our EETO algorithm performs better in terms of both long-term average energy consumption and queue length. Compared with the LCA algorithm, the energy consumption of LCA is smaller than that of our proposed EETO algorithm, but we can see that the queue length of the LCA algorithm is much higher than that of our EETO algorithm especially when the maximum CPU cycle frequency of IoT devices is at a lower level. As for the ICSOC-19 algorithm, we can see that the queue length is close to our algorithm with the increase of maximum CPU-cycle frequency, but we can find that the energy consumption of the ICSOC-19 is higher than EETO algorithm. Compared with DOLA-22 and CCPM-19 algorithms, we can find that the queue length is the lowest. However, we can also find that compared with these two algorithms, the EETO algorithm exhibits superior performance in terms of energy consumption. In conclusion, together with Fig. 6(a) and (b), it is shown that the EETO algorithm outperforms the SDTO-18 algorithm in both energy consumption and queue length. Compared with the LCA algorithm, our EETO algorithm reduces energy consumption while ensuring queue stability. Compared with ICSOC-19, DOLA-22, and CCPM-19 algorithms, the EETO algorithm reduces energy consumption with a little sacrifice of queue length.

Next, we further depict the total energy consumption and queue length with different time slots. As shown in Fig. 7, we illustrate the change in total queue length and energy consumption over 5000 time slots with the maximum CPU cycle frequency of IoT devices being 1 GHz (i.e., $\beta = 1.0$). Compared with the LCA algorithm, and SDTO-18 algorithm, our proposed EETO algorithm can maintain queue stability. As for the SDTO-18 algorithm, the transmission and computation resources of the ES are limited, and a large number of computation tasks are transmitted, which causes a large backlog of offloading queues and queues in MEC servers. Regarding the LCA algorithm, the reason is due to the limited local calculation ability. IoT devices can not process all the computation tasks locally. Compared with the ICSOC-19, DOLA-22, and CCPM-19 algorithms, the three algorithms can ensure queue stability. However, from Fig. 7(b) we can see that in terms of energy consumption, the EETO algorithm is slightly smaller than the ICSOC-19 algorithm and significantly smaller than the DOLA-22 and CCPM-19 algorithms. Therefore, EETO algorithm can reduce the energy consumed by the system with a little sacrifice of queue length.

V. Related Work

Computation tasks can be transmitted to ESs for calculation because of the IoT device's limited battery and computation capacity [22]. By formulating a suitable resource allocation and computation task offloading strategy, such problems can be effectively addressed. How many computation tasks to offload is the core of the offloading strategy that needs to be determined. Offloading decisions are usually divided into binary offloading decisions and partial offloading decisions. Mao et al. [13] designed and proposed a binary offloading algorithm, which optimized the allocation of transmission and computation resources in MEC systems simultaneously. This algorithm achieved the balance of latency and energy consumption performance. Huang et al. [23] established energy efficient task scheduling model for sensor hubs. Further, the authors proposed a multi-queue scheduling method to attack the challenge of energy constraints. The effectiveness of this approach was verified through simulation experiments based on real-life data. Tang et al. [11] investigated ES selection taking into account delay-sensitive and indivisible computation tasks and ES load dynamics. Their optimization goal was to decrease the expected costs of continuous time slots. Guo et al. [24] investigated the computation offloading problem in ultradense IoT networks with a set of base stations. Then, they presented an iterative searching-based computation offloading scheme to achieve the solution that jointly optimized computation offloading, transmit power allocation, and computational frequency scaling.

Besides binary offloading, there are also a lot of works on partial offloading decisions. Hu et al. [25] studied the balance between latency and energy efficiency of devices in MEC. Their decisions contain how many computation tasks to calculate by the local CPU and how many computation tasks to be offloaded to obtain maximum energy efficiency. Huang et al. [26] jointly considered the fairness and throughput with the goal of keeping the buffer from being congested and maximizing the utility. Moreover, a named ACCRA algorithm was

presented to cope with the problem. Theoretical analysis and simulation experiments demonstrated the effectiveness of their proposed algorithm. Guo et al. [27] investigated the problem of resource scheduling and offloading in an energy harvesting scenario, jointly considering IoT devices and SBS. Minimizing the consumption of energy subjected to QoS constraint was their optimization objective. Duan et al. [28] studied the mobility-aware online task offloading problem and aimed to minimize the total computation costs. To effectively solve this problem, the original problem was transformed and decoupled into two sub-problems. Then, the authors proposed the LSTM-based algorithm and Dueling Double DQN-based algorithm to solve these two sub-problems.

The lack of spectrum resources has become a factor that limits the efficient transmission of devices due to the explosive growth of IoT devices [29]. Mao et al. [30] considered FDMA and TDMA modes and studied the online offloading policy. Their optimization goal is to balance latency and energy efficiency in MEC systems that support wireless power. Huang et al. [9] aimed to maximize computation rate which integrated task offloading decisions and resource allocations. They adopted TDMA for wireless devices to offload tasks to the access point and proposed a DROO algorithm via deep reinforcement learning to address this problem. Lyu et al. [31] considered maximizing the long-term mobile edge caching benefits in large-scale WiFi systems. By conducting intensive spatiotemporal analysis of WiFi traffic consumption, the TEG algorithm was designed to solve the problem of maximizing long-term cache benefit. Huang et al. [32] combined MEC and federated learning to solve the problem of resource shortage and privacy leakage. The investigated problem was modeled as a two-stage Stackelberg game. Then, the authors proposed a game-based incentive mechanism algorithm to address the problem.

NOMA is considered to be a promising multiple access scheme because it can support ultra-dense connections, ultra-high throughput, and ultra-low latency. Pan et al. [33] adopted NOMA in MEC systems and studied tasks upload and results download simultaneously. Then, they concluded that the MEC system based on NOMA could consume less energy than other transmission protocols. Wu et al. [34] studied NOMA-aided federated learning. In order to conduct model aggregation, a group of devices from the NOMA cluster transmitted the data of the trained model to ES. They aimed to decrease the cost by accounting for the federated learning convergence latency and total energy consumed by the devices. Dai et al. [35] investigated the problem of latency in a NOMA-based MEC scenario. The study adjusts the allocation of resources in both computation and communication to achieve the minimum total latency.

VI. Conclusion

The NOMA-aided dynamic task offloading problem is studied which involves task scheduling and resource allocation for IoT in MEC systems. Under the premise of queue stability, optimizing the system's energy consumption is our objective. The initial stochastic problem is converted to a deterministic optimization problem based on stochastic optimization technology. Then, the deterministic problem is divided into four sub-problems. The EETO algorithm for solving these four sub-problems is proposed, which does not rely on prior statistical knowledge relevant to channel conditions or task arrival process. Through theoretical analysis, the EETO algorithm can balance the system's energy consumption and queue stability. Experimental results validate that our EETO algorithm can optimize energy consumption while effectively guaranteeing the system's performance. In our future work, we will further go deeply into joint SBS selection and channel selection in NOMA-aided scenarios.

References

[1] P. Lai et al., "Dynamic user allocation in stochastic mobile edge computing systems," *IEEE Trans. Serv. Comput.*, vol. 15, no. 5, pp. 2699–2712, Sep./Oct. 2022.

[2] S. Rajendran and Z. Sun, "RF impairment model-based IoT physical-layer identification for enhanced domain generalization," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1285–1299, 2022.

[3] P. Zhang, H. Jin, H. Dong, W. Song, and A. Bouguettaya, "Privacy-preserving QoS forecasting in mobile edge environments," *IEEE Trans. Serv. Comput.*, vol. 15, no. 2, pp. 1103–1117, Mar./Apr. 2022.

[4] Q. Su, Q. Zhang, W. Li, and X. Zhang, "Primal-dual-based computation offloading method for energy-aware cloud-edge collaboration," *IEEE Trans. Mobile Comput.*, vol. 23, no. 2, pp. 1534–1549, Feb. 2024.

[5] Y. Chen, K. Li, Y. Wu, J. Huang, and L. Zhao, "Energy efficient task offloading and resource allocation in air-ground integrated MEC systems: A distributed online approach," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2023.3346431.

[6] H. Jin, P. Zhang, H. Dong, Y. Zhu, and A. Bouguettaya, "Privacy-aware forecasting of quality of service in mobile edge computing," *IEEE Trans. Serv. Comput.*, vol. 16, no. 1, pp. 478–492, Jan./Feb. 2023.

[7] A. Mohajer, M. Sam Daliri, A. Mirzaei, A. Ziaeddini, M. Nabipour, and M. Bavaghar, "Heterogeneous computational resource allocation for NOMA: Toward green mobile edge-computing systems," *IEEE Trans. Serv. Comput.*, vol. 16, no. 2, pp. 1225–1238, Mar./Apr. 2023.

[8] Y. Chen, J. Zhao, J. Hu, S. Wan, and J. Huang, "Distributed task offloading and resource purchasing in noma-enabled mobile edge computing: Hierarchical game theoretical approaches," *ACM Trans. Embedded Comput. Syst.*, vol. 23, pp. 1–28, 2023.

[9] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, Nov. 2020.

[10] H. Hu, W. Song, Q. Wang, R. Q. Hu, and H. Zhu, "Energy efficiency and delay tradeoff in an MEC-enabled mobile IoT network," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15942–15956, Sep. 2022.

[11] M. Tang and V. W. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1985–1997, Jun. 2022.

[12] W. Lin, T. Huang, X. Li, F. Shi, X. Wang, and C.-H. Hsu, "Energy-efficient computation offloading for UAV-assisted MEC: A two-stage optimization scheme," *ACM Trans. Internet Technol.*, vol. 22, no. 1, pp. 1–23, Oct. 2021.

[13] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[14] M. J. Neely and J. Michael, "Stochastic network optimization with application to communication and queueing systems," *Synth. Lectures Commun. Netw.*, vol. 3, no. 1, 2010, Art. no. 211.

[15] J. Huang, M. Wang, Y. Wu, Y. Chen, and X. Shen, "Distributed offloading in overlapping areas of mobile-edge computing for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13837–13847, Aug. 2022.

[16] Z. Tong, J. Cai, J. Mei, K. Li, and K. Li, "Dynamic energy-saving offloading strategy guided by Lyapunov optimization for IoT devices," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19903–19915, Oct. 2022.

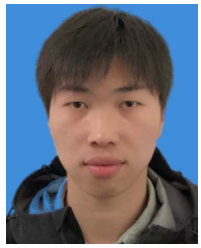
[17] B. Liu, C. Liu, and M. Peng, "Resource allocation for energy-efficient MEC in NOMA-enabled massive IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1015–1027, Apr. 2021.

[18] 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.

[19] P. Lai et al., *Edge User Allocation With Dynamic Quality of Service*. Berlin, Germany: Springer, 2019, pp. 86–101.

[20] J. Huang, M. Wang, Y. Wu, Y. Chen, and X. Shen, "Distributed offloading in overlapping areas of mobile edge computing for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13837–13847, Aug. 2022.
[21] H. Zeng, X. Zhu, Y. Jiang, Z. Wei, and T. Wang, "A green coordinated multi-cell NOMA system with fuzzy logic based multi-criterion user mode selection and resource allocation," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 480–495, Jun. 2019.
[22] S. Duan et al., "Distributed artificial intelligence empowered by end-edge-cloud computing: A survey," *IEEE Commun. Surv. Tut.*, vol. 25, no. 1, pp. 591–624, First Quarter 2023.
[23] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chin. J. Electron.*, vol. 29, no. 2, pp. 242–247, 2020.
[24] H. Guo, J. Zhang, J. Liu, and H. Zhang, "Energy-aware computation offloading and transmit power allocation in ultradense IoT networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4317–4329, Jun. 2019.
[25] H. Hu, W. Song, Q. Wang, R. Q. Hu, and H. Zhu, "Energy efficiency and delay tradeoff in an MEC-enabled mobile IoT network," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15942–15956, Sep. 2022.
[26] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1964–1973, Feb. 2022.
[27] M. Guo, W. Wang, X. Huang, Y. Chen, L. Zhang, and L. Chen, "Lyapunov-based partial computation offloading for multiple mobile devices enabled by harvested energy in MEC," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 9025–9035, Jun. 2022.
[28] S. Duan et al., "MOTO: Mobility-aware online task offloading with adaptive load balancing in small-cell MEC," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 645–659, Jan. 2024.
[29] G. Cui, Q. He, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Location privacy protection via delocalization in 5G mobile edge computing environment," *IEEE Trans. Serv. Comput.*, vol. 16, no. 1, pp. 412–423, Jan./Feb. 2023.
[30] S. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Energy efficiency and delay tradeoff for wireless powered mobile-edge computing systems with multi-access schemes," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1855–1867, Mar. 2020.
[31] F. Lyu et al., "Lead: Large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics," *IEEE Trans. Mobile Comput.*, vol. 20, no. 8, pp. 2607–2623, Aug. 2021.
[32] J. Huang et al., "Incentive mechanism design of federated learning for recommendation systems in MEC," *IEEE Trans. Consum. Electron.*, to be published, doi: 10.1109/TCE.2023.3342187.
[33] Y. Pan, M. Chen, Z. Yang, N. Huang, and M. Shikh-Bahaei, "Energy-efficient NOMA-based mobile edge computing offloading," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 310–313, Feb. 2019.
[34] Y. Wu, Y. Song, T. Wang, L. Qian, and T. Q. S. Quek, "Non-orthogonal multiple access assisted federated learning via wireless power transfer: A cost-efficient approach," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2853–2869, Apr. 2022.
[35] Y. Dai, M. Sheng, J. Liu, N. Cheng, and X. Shen, "Resource allocation for low-latency mobile edge computation offloading in NOMA networks," in *Proc. IEEE Glob. Commun. Conf.*, 2018, pp. 1–6.

Ying Chen (Senior Member, IEEE) received the PhD degree in computer science and technology from Tsinghua University, Beijing, China, in 2017. She was a joint PhD student with the University of Waterloo, Waterloo, ON, Canada from 2016 to 2017. She is a professor with the Computer School, Beijing Information Science and Technology University, Beijing. Her current research interests include Internet of Things, mobile edge computing, wireless networks and communications, machine learning, etc. She is the recipient of the Best Paper Award with IEEE SmartIoT 2019, the 2016 Google PhD Fellowship Award, and the 2014 Google Anita Borg Award, 2022 OUTSTANDING CONTRIBUTION AWARD in 18th EAI CollaborateCom, respectively. She serves/served the leading guest editor of Springer JCC, TPC member of IEEE HPCC, and PC member of IEEE Cloud, CollaborateCom, IEEE CPSCom, CSS, etc. She is also the Reviewer of several journals such as the *IEEE Wireless Communications Magazine*, *IEEE Transactions on Dependable and Secure Computing*, *IEEE Internet of Things Journal*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, and *IEEE Transactions on Services Computing*.



Jiajie Xu is currently working toward the MEng degree in computer science and technology, with the Beijing Information Science and Technology University, China. His current research interests include edge computing, stochastic optimization theory, and deep reinforcement learning.

Yuan Wu (Senior Member, IEEE) received the PhD degree in electronic and computer engineering from the Hong Kong University of Science and Technology, in 2010. He is currently an associate professor with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao, China, and also with the Department of Computer and Information Science, University of Macau. His research interests include resource management for wireless networks, green communications and computing, edge computing and edge intelligence, and energy informatics. He received the Best Paper Award from the IEEE ICC'2016, IEEE TCGCC'2017, IWCMC'2021, and WCNC'2023. He serves/served as the Track/Symposium Co-Chair for IEEE VTC'2017-Fall, VTC'2021-Spring, VTC'2022-Spring, ICCC'2023, and GLOBECOM'2024. He is currently on the editorial board of *IEEE Transactions on Vehicular Technology*, *IEEE Transactions on Network Science and Engineering*, and *IEEE Internet of Things Journal*.

Jie Gao (Senior Member, IEEE) received the MSc and PhD degrees in electrical engineering from the University of Alberta, Edmonton, AB, Canada, in 2009 and 2014, respectively. He was a postdoctoral fellow with Toronto Metropolitan (formerly Ryerson) University, Toronto, ON, from 2017 to 2019 and a research associate with the University of Waterloo, Waterloo, ON, from 2019 to 2020. He was an assistant professor with the Department of Electrical and Computer Engineering, Marquette University, Milwaukee, WI, USA, from 2020 to 2022 and is currently an assistant professor with the School of Information Technology, Carleton University, Ottawa, ON. He is research interests include machine learning for communications and networking, cloud and multi-access edge computing, Internet of Things (IoT) and industrial IoT solutions, and B5G/6 G networks in general.

Lian Zhao (Fellow, IEEE) received the PhD degree from the Department of Electrical and Computer Engineering (ELCE), University of Waterloo, Canada, in 2002. She joined the Department of Electrical and Computer Engineering with Toronto Metropolitan University (formerly Ryerson University), Canada, in 2003. Her research interests are in the areas of wireless communications, resource management, mobile edge computing, caching and communications, and IoV networks. She has been an IEEE Communication Society (ComSoc) and IEEE Vehicular Technology (VTS) Distinguished Lecturer (DL); received the Best Land Transportation Paper Award from IEEE Vehicular Technology Society in 2016, Top 15 Editor Award in 2016 for *IEEE Transactions on Vehicular Technology*, Best Paper Award from the 2013 International Conference on Wireless Communications and Signal Processing (WCSP), and the Canada Foundation for Innovation (CFI) New Opportunity Research Award in 2005. She has been serving as an editor for *IEEE Transactions on Wireless Communications*, *IEEE Internet of Things Journal*, and *IEEE Transactions on Vehicular Technology* (2013–2021).