Peer-to-Peer Networking and Applications (2023) 16:2023–2038
https://doi.org/10.1007/s12083-023-01522-9

RIS-assisted device-edge collaborative edge computing for industrial applications

Mian Guo[1] · **Chengyuan Xu**[1] · **Mithun Mukherjee**[2]

Received: 11 April 2023 / Accepted: 20 June 2023 / Published online: 29 June 2023

Abstract
In the Industrial Internet of Things (IIoT), a significant amount of perceived data is generated from massive IoT devices, which requires timely computing for value maximization. Multi-access edge computing (MEC), which deploys computing nodes close to the data source, is a promising computing paradigm for IIoT applications. However, due to the limited computation resource, it is challenging for edge nodes to provide a low delay to massive data. In addition, the wireless transmission environment varies with IoT devices over time. Some data even cannot be uploaded to the edge server due to the worse link quality. Reconfigurable intelligent surface (RIS), which deploys passive reflecting elements between end users and base station to reflect wireless signals, is a new technique for changing the wireless transmission performance via reconfiguring the phase shift of RIS. It is beneficial to apply RIS in MEC for reducing transmission delay and achieving green edge computing. This paper considers a RIS-assisted device-edge collaborative MEC for industrial applications. We propose to minimize the energy consumption of IoT devices constrained to the delay requirements via jointly optimizing the offloading decisions between end and edge computing nodes, the phase shift of RIS, CPU resource allocation of edge server, and transmission power of IoT devices. A distributed and cooperative scheme, called RIS-assisted DAEM, which includes the DAECO and DCEM algorithms for CO and PORA subproblems, respectively, is proposed to solve the formulated problem. The simulation results have illustrated the efficiency of the proposal for energy consumption reduction constrained to the delay requirements.

Keywords Edge computing · Computation offloading · Reconfigurable intelligent surface (RIS) · Device-edge collaboration

1 Introduction

In the industrial Internet of Things (IIoT), most productions/machines generate much-perceived data, which requires high-performance computing and tolerates low delay to support typical IIoT applications, such as remote maintenance, intelligent factories, smart logistics, and so on [1]. Cloud computing has witnessed success in high-performance computing for data analysis and mining. However, with the ever-increasing data generated from various IIoT applications, a massive amount of data bursts into the cloud center, which has seriously exhausted the bandwidth resource of the communication network from data source to cloud center [2]. In addition, the cloud computing paradigm exposes data to public networks and computing centers, leading to intolerable long network transmission delays and rising security and privacy issues, respectively.

In recent years, multi-access edge computing (MEC) has been considered a promising computing paradigm for applications demanding low delay and high security. In MEC, edge servers with cloud-like computing capability are deployed at the network edge close to the data source. In addition, Internet of Things (IoT) devices (e.g., machines with sensors and communication modules) are also endowed with some computing capability. Thus, data

This is part of the Topical Collection: *1- Track on Networking and Applications*

✉ Mian Guo
mianguo@gpnu.edu.cn

Chengyuan Xu
15937283669@163.com

Mithun Mukherjee
m.mukherjee@ieee.org

[1] School of Electronics and Information, Guangdong Polytechnic Normal University, Guangzhou 510660, China

[2] School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China

could be processed locally at IoT devices or offloaded to an edge server for computing, which would reduce network transmission delay and improve data security and privacy.

However, since most productions/machines work as IoT devices in an IIoT environment, the competition for shared bandwidth and computing resources is still very intense in MEC. In addition, due to the dynamic nature of the wireless transmission environment, some data may be unable to be offloaded from the data source to the edge server for computing if the wireless link is blocked or worse. Furthermore, the battery energy of an IoT device is limited due to its small volume size. Therefore, it is desired to study how to efficiently utilize the computation resource of IoT devices and edge servers as well as the precious wireless resource to achieve green MEC for IIoT applications.

Recently, reconfigurable intelligent surface (RIS) has appeared as a new emerging wireless technology for energy efficiency and transmission performance improvement [3, 4]. In RIS, several passive reflecting elements are deployed to establish extra links between wireless users and the base station (BS) [5]. The RIS elements could be reconfigured according to the surrounding environment, such that high beamforming gain between users and BS could be achieved [6]. In addition, the passive RIS elements just reflect the wireless signals; no extra transmission power is required for IoT devices. It is beneficial to apply RIS in edge computing for reducing transmission delay while improving energy efficiency. However, RIS is presently in its initiate stage, how to merge RIS into edge computing desires further study.

Motivated by the attractive characteristics of RIS, this paper studies a RIS-assisted device-edge collaborative edge computing problem in an industrial environment, aiming at reducing the energy consumption of IoT devices constrained to the delay requirements of computation tasks from IIoT applications by jointly optimizing (a) the offloading decisions, e.g., local computing at IoT devices, or, offloading to the edge server via the RIS-assisted wireless network for edge computing; (b) the phase shift in RIS for high beamforming gain; (c) the computation resource allocation for offloaded tasks at edge server; (d) the transmission power of IoT devices for task offloading.

The main contributions are summarized as follows.

- A joint computation offloading and resource allocation problem (called JCORA) for energy efficiency in a RIS-assisted device-edge collaborative edge computing IIoT environment is formulated. In order to find out a low complex scheme for addressing such non-convex NP-hard problem, the problem is further decomposed into two concatenated subproblems, computation offloading (CO) and the joint RIS phase beamforming optimization and computation resource allocation (PORA).
- A distributed and cooperative scheme, called RIS-assisted delay-aware energy minimization computation offloading (RIS-assisted DAEM), which consists of delay-aware energy consumption minimization offloading (DAECO) for the CO subproblem and delay-constrained transmission-energy-minimization resource optimization (DCEM) for the PORA subproblem, is proposed to address the JCORA problem. By joining DAECO and DCEM in RIS-assisted DAEM, optimum offloading decisions, RIS phase shift, computation resource allocation strategy, as well as transmission power are found.
- The simulations have illustrated that, RIS-assisted DAEM can significantly improve the energy efficiency constrained to the end-to-end delay requirements of computation tasks in comparison with benchmarked algorithms. In addition, RIS-assisted DAEM also outperforms benchmarked algorithms in delay guarantee capability.

The remainder of this paper is organized as follows. Section 2 describes the related work. Section 3 presents the system, delay and energy consumption models. In Section 4, the problem is formulated and addressed in detail. We provide the performance evaluation of the proposal through computer simulations in Section 5, and conclude the paper in Section 6.

2 Related work

Edge computing has attracted significant attentions in recent years [7, 8]. By deploying cloud-like computing nodes (e.g., edge server, end-device) in the network edge close to data source, edge computing could explicitly reduce network transmission delay of computation tasks. Since the computation resource of an edge node is limited in comparison with that of cloud computing, computation offloading problem such as where to process the tasks and how to upload the offloaded tasks arise with the appearance of edge computing.

A number of edge computing algorithms have been proposed for achieving different objectives.

Edge computing for energy efficiency. Energy-aware edge computing has been studied for various types of networks with various methods. Zhou et al. has studied the

energy-efficient workload offloading problem in vehicular networks [9]. The authors have proposed a low-complex distributed solution based on consensus alternating direction method of multipliers to address the problem for energy saving. Zhang et al. studied the energy-efficient computation offloading for mobile edge computing in 5 G heterogeneous networks [10]. By jointly optimizing offloading and radio resource allocation, the minimal energy consumption under the latency constraints is achieved in [10]. Song et al. considered the energy-efficient computation offloading problem for terrestrial-satellite IoT [11]. They decomposed the computation offloading problem into two layered subproblems and solved separately. Then an energy-efficient computation offloading and resource allocation algorithm based on the solutions of the two subproblems are proposed. The joint optimization of computation offloading and resource allocation for energy efficiency in a dynamic multiuser MEC system is considered in [12]. The problem is formulated as a mixed-integer nonlinear programming (MINLP) problem and solved via a value iteration-based reinforcement learning method. Wang et al. proposed a task offloading and resource allocation mechanism, which considered the clock frequency configuration, transmission power allocation, channel rate scheduling and offloading strategy selection, to achieve energy-efficient offloading performance in MEC [13].

Edge computing for delay minimization. Since one of the most important goal of edge computing is to reduce the delay of computation tasks, many proposals have focused on the issues of delay minimization in edge computing. Yi et al. studied a MEC framework with multiuser computation offloading and transmission scheduling for delay-sensitive applications [14]. By considering tradeoffs between local and edge computing, wireless features and non-cooperative game interactions among mobile users, a joint computation offloading and transmission scheduling as well as pricing rule problem for such MEC framework is formulated and solved in [14]. Kuang et al. studied the joint problem of cooperative computation task offloading and resource assignment in MEC [15]. The objective of minimizing the latency while guaranteeing the constraint of transmission power, energy consumption and CPU cycle frequency is achieved by formulating the optimization problem as a nonconvex mixed-integer problem and solving via a joint iterative algorithm. The delay-aware and energy-efficient computation offloading in a dynamic MEC with multiple edge servers has been studied in [16]. The authors proposed an end-to-end deep reinforcement learning (DRL) approach to address the computation offloading problem for maximizing the completed tasks before their respective deadlines and minimizing energy consumption. Shahryari et al. also proposed an energy-efficient and delay-guaranteed computation offloading method for fog-based IoT networks [17]. Differently, Liu et al. studied the joint optimization of energy and delay for computation offloading in cloudlet-assisted mobile cloud computing [18]. Chen et al. has studied the problem of joint computation offloading and unmanned aerial vehicle (UAV) deployment for average task response time minimization [19], where a two-layer joint optimization method, called PSO-GA-G, is proposed to address the problem.

RIS-assisted wireless communication. RIS, also termed as intelligent reflecting surface (IRS) [6, 20, 21], is a promising new solution to energy-efficiently improve the wireless transmission performance towards fifth-generation (5 G) and sixth-generation (6 G) networks. In RIS, a number of low-cost reconfigurable passive elements is deployed. By smartly adjusting the phase shift of all passive elements in RIS, the reflected signals can change the wireless propagation environment. For example, by adjusting the reflecting beamforming coherently with the signals from other paths, the received signal power at the receiver would be enhanced. Since RIS has attractive advantages for wireless performance improvement, RIS-assisted wireless communication has been widely studied recently. Wei et al. studied the channel estimation issue for RIS assisted wireless communications [5, 22]. Wu et al. studied the joint RIS phase shift optimization and wireless powered non-orthogonal multiple access (NOMA) resource allocation problem, the aim is to maximize the sum throughput of all wireless powered devices [23]. The impact of spatial channel correlation on the outage probability of IRS-assisted single-input single-output (SISO) communication systems has been studied in [24]. A downlink RIS-assisted multiple-input multiple-output (MIMO) wireless communication system that comprising three communication links of Rician channel, e.g., links from BS to RIS, RIS to user, and BS to user has been studied in [3], where an optimal transmit covariance matrix at BS and diagonal phase-shifting matrix at RIS have been explored to maximize the achievable ergodic rate with statistical channel state information at BS. Al-Hilo et al. designed a RIS-assisted UAV method for timely data collection in IoT networks [25]. Different from other studies that just focus on the reflection function of RIS, Zuo et al. proposed a joint design for simultaneously transmitting and reflecting (STAR) RIS assisted NOMA system to maximize the achievable sum rate [4]. Sankar et al. considered a hybrid RIS comprising of active

and passive elements to aid an integrated sensing and communication system [26]. The authors jointly designed transmit beamformers and RIS coefficients to maximize the worst-case target illumination power while ensuring a desired signal-to-interference-plus-noise ratio for communication links and constraining the RIS noise power due to the active elements. An effective iterative algorithm for solving the problem of joint phase-shifts of the RIS and the resource allocation of the relays in RIS-assisted multi-hop MEC network for network throughput optimization has been designed in [27]. The joint trajectory-task-cache optimization with phase-shift design has been considered by placing a RIS between UAV, which works as MEC server, and ground terminals (GTs), for performance improving on mobile computing [28]. Bai et al. has used a block coordinate descent (BCD) technique to solve the joint computation offloading and RIS phase shift design problem in RIS aided MEC for latency minimization [29]. The joint optimization of the CPU frequencies of the smart terminals (STs), the offloading schedule, the RIS phase shifts, and the receive beamformers of the BS for minimizing the energy consumption of the STs in RIS-assisted MEC has been studied in [30].

Different from the existing works, this paper studies the RIS-assisted device-edge collaborative edge computing for industrial applications, where we jointly optimize the computation offloading and phase shift of RIS as well as computation resource allocation of edge servers for minimizing the energy consumption of IoT devices constrained to the delay requirements of industrial tasks. Furthermore, since the offloading decisions, phase shift of RIS, computation resource allocation, and transmission power of IoT devices affect each other, we decompose the joint computation offloading and multiple resource allocation problem into two subproblems and solve with distinct algorithms. Then, a distributed and cooperative scheme, which iteratively combines the up-to-date solutions of the above two subproblems, is designed to obtain one-shot solution for achieving the goal of energy consumption minimization constrained to the delay requirements of industrial tasks. In addition, different from existing works that generally transform the RIS phase shift optimization problem into a convex semidefinite program and then apply semidefinite relaxation (SDR) solver [31] to obtain the phase shift [6, 26, 32], which is off-line and time-consuming, we design a low-complex iterative algorithm to find out the suboptimal phase shift for multiple users.

3 Model formulation

3.1 System model

As illustrated in Fig. 1, this paper considers a RIS-assisted MEC system in industrial environments. The system consists of a base station endowed with an edge server located at the center of the wireless network, a number of IoT devices (e.g., machines, industrial terminals) randomly located in the network. The edge server can process multiple tasks in parallel via virtualization technologies [33], while each IoT device could only serve one task per time. There is a RIS deploying in a location between IoT devices and the base station for enhancing the communication efficiency.

The IoT devices generate industrial computation tasks. For each task, the corresponding IoT device has to decide where to process the task, e.g., local computing, or, offloading to the edge server for edge computing. If the decision is offloading, then, the task should be uploaded to the edge server via the RIS-assisted wireless network before its processing at the edge server; otherwise, the task will be executed locally. In addition, for the offloaded tasks, there are another two types of

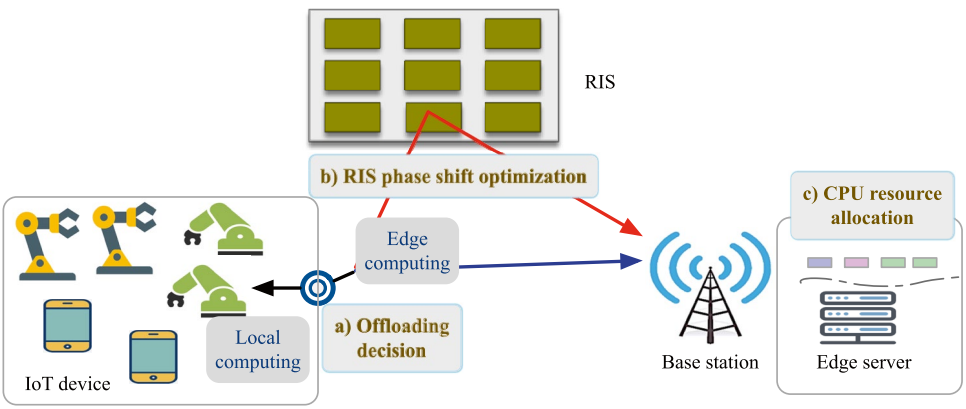


Fig. 1 Model of a RIS-assisted MEC system

Table 1 Summary of notations

Symbols	Definition
M	The number of IoT devices
N	Reflecting elements in the RIS
h^{I}	The channel coefficient from IoT devices to RIS
h^{D}	The channel coefficient from IoT devices to BS
g^{H}	The channel coefficient from RIS to BS
p_m	The transmission power of IoT device m
B	The bandwidth of the wireless link
σ_0	The noise power of the wireless network
I_m	Offloading decision variable
f^{O}	The computing resource of the edge server
f_m^{O}	The computing rate allocated to task m
f_m^{L}	The computing resource of IoT device m
Λ	The reflection coefficient matrix in the RIS
θ	The phase shift in the RIS
S_m	The task size (in bits) of IoT device m
W_m	Task size (in CPU cycles)
$E_m^{\mathrm{O,\,Tx}}$	The energy consumption for data upload
E_m^{L}	The energy consumption for local computing
A^H	The conjugate transpose of matrix A
$\mathcal{M}$	The IoT device set
$\mathcal{M}^{\mathrm{O}}$	The IoT device set for edge computing
$\mathcal{M}^{\mathrm{L}}$	The IoT device set for local computing
$\mathcal{N}$	The set of passive reflecting elements in RIS

decisions are further needed to make. The first is the RIS phase beamforming optimization, which includes the phase shift of passive reflecting elements in RIS and the transmission power of the offloaded IoT devices, for energy-efficiently improving the task upload rate. The second type of decision is the computation resource allocation at edge server, which decides how to allocate the computation resource among offloaded tasks for quality of service (QoS) provisioning.

Let $\mathcal{M} = \{1, 2, \ldots, M\}$ be the IoT device set in the system. Assuming the passive reflecting elements equipped in the RIS is N. The channel coefficient from IoT device $m \in \mathcal{M}$ to RIS, from IoT device m directly to the BS, and from the RIS to the BS are represented by $h_m^I \in \mathbb{C}^{N\times 1}$, $h_m^D \in \mathbb{C}^{1\times 1}$ and $g^H \in \mathbb{C}^{1\times N}$, respectively.

Let S_m represent the task size (in bits) generated from IoT device $m \in \mathcal{M}$, the corresponding computation size is denoted as W_m (in CPU cycles).

Let $\mathcal{I} = (I_1, I_2, \ldots, I_m, \ldots, I_M)$ be the offloading decision vector, where $I_m = 1$ if the task generated from IoT device m is offloaded to edge server, $I_m = 0$ otherwise. Let $\mathcal{M}^{\mathrm{O}} \subset \mathcal{M}$ represent the IoT device set that offloads the tasks to edge server, and $\mathcal{M}^{\mathrm{L}} \subset \mathcal{M}$ is the IoT device set that determines to local computing.

The summary of notations is listed in Table 1.

3.2 RIS-assisted edge computing model

When a task from IoT device m is offloaded to the edge server for computing, it will experience two types of delays, upload delay $D_m^{\mathrm{O,\,Tx}}$ from data source to the edge server and the computation delay $D_m^{\mathrm{O,\,CPU}}$ at the edge server. Thus, the edge computation delay of the task is expressed by

$$D_m^{\mathrm{O}} = D_m^{\mathrm{O,\,Tx}} + D_m^{\mathrm{O,\,CPU}}. \tag{1}$$

The upload delay is derived by

$$D_m^{\mathrm{O,\,Tx}} = \frac{S_m}{R_m}, \tag{2}$$

where R_m is the transmission rate from IoT device m to the edge server (we ignore the delay from the BS to the edge server), which is expressed by

$$R_m = B\log_2\left(1 + \frac{p_m|g^{\mathrm{H}}\Lambda h_m^{\mathrm{I}} + h_m^{\mathrm{D}}|^2}{\sigma_0}\right), \tag{3}$$

where B is the bandwidth, p_m is the transmission power, $\Lambda = \mathrm{diag}(e^{j\theta_1}, e^{j\theta_2}, \ldots, e^{j\theta_n}, \ldots, e^{j\theta_N})$ is the reflection coefficient matrix, where $\theta_n \in [0, 2\pi]$ is the phase shift of the nth passive reflecting element in RIS, and σ_0 is the additive Gaussian noise power. Define $\theta = (\theta_1, \theta_2, \ldots, \theta_N)$ as the phase shift of the passive reflecting elements in the RIS.

As seen in Eq. (3), with RIS, the signal-to-noise ration (SNR) of the upload transmission is changed to the complex superposition of the reflect-path (e.g., $g^{\mathrm{H}}\Lambda h_m^{\mathrm{I}}$) and direct-path (e.g., h_m^{D}) signals. In this case, the phase shift (e.g, Λ) has important impact on the superposition signal. For example, some range of phase shift could enhance the signal, while some others may suppress the direct-path signal at the receiver.

The computation delay at the edge server is derived by

$$D_m^{\mathrm{O,CPU}} = \frac{W_m}{f_m^{\mathrm{O}}}, \tag{4}$$

where f_m^{O} is the CPU rate of the edge server allocated to the task, which is constrained by

$$\sum_{m\in\mathcal{M}} I_m f_m^{\mathrm{O}} \le f^{\mathrm{O}}. \tag{5}$$

In this paper, we mainly consider the energy consumption of IoT devices, due to its batter volume constraint and difficult to harvest energy. Therefore, the energy consumption for task offloading is defined as

$$E_m^{\mathrm{O}} = E_m^{\mathrm{O,\,Tx}}. \tag{6}$$

Since the passive reflecting element in the RIS does not consume energy [34], the energy consumed for data upload is mainly determined by its transmission power and transmission duration (equivalently, upload delay) [35]. Accordingly, the energy consumption for data upload could be expressed by

$$E_m^{\mathrm{O,\,Tx}} = p_m D_m^{\mathrm{O,\,Tx}}. \tag{7}$$

3.3 Local computing model

When a task from IoT device m is determined to local computing, it will only experience the data computation delay D_m^{L} at local IoT device. That is,

$$D_m^{\mathrm{L}} = \frac{W_m}{f_m^{\mathrm{L}}}. \tag{8}$$

Therefore, the energy consumption for local computing is just the energy consumed for data computing. That is,

$$E_m^{\mathrm{L}} = \beta W_m f_m^{\mathrm{L}} = \beta D_m^{\mathrm{L}} (f_m^{\mathrm{L}})^2, \tag{9}$$

where β is the energy factor at local IoT device.

3.4 Problem formulation

The end-to-end delay of a task generated from IoT device m could be uniformly expressed as

$$D_m = (1 - I_m) D_m^{\mathrm{L}} + I_m D_m^{\mathrm{O}}. \tag{10}$$

Similarly, the energy consumption for processing the task from IoT device m could be uniformly expressed as

$$E_m = (1 - I_m) E_m^{\mathrm{L}} + I_m E_m^{\mathrm{O}}. \tag{11}$$

The total energy consumption of all tasks from all IoT devices is defined as

$$E = \sum_{m \in \mathcal{M}} E_m. \tag{12}$$

In this paper, we aim at minimizing the total energy consumption constrained to tasks' delay requirements by jointly optimizing the offloading decision $\mathcal{I}$, RIS phase beamforming Λ, transmission power $\mathtt{p} \triangleq \{p_m : \forall m \in \mathcal{M}^{\mathrm{O}}\}$, and computation resource allocation $\mathtt{f} \triangleq \{f_m^{\mathrm{O}} : \forall m \in \mathcal{M}^{\mathrm{O}}\}$ for offloaded tasks. We call the above problem as the JCORA problem, and formulate it as

$$\mathrm{JCORA}: \min_{\mathcal{I},\Lambda,\mathtt{p},\mathtt{f}} \; E \tag{13a}$$

$$\text{s.t.} \quad D_m \le d_m^{\mathrm{Th}}, \forall m \in \mathcal{M} \tag{13b}$$

$$I_m \in \{0, 1\}, \forall m \in \mathcal{M} \tag{13c}$$

$$\mathcal{M}^{\mathrm{O}} \cup \mathcal{M}^{\mathrm{L}} \subseteq \mathcal{M} \tag{13d}$$

$$0 \le \theta_n \le 2\pi, \forall n \in \mathcal{N} \tag{13e}$$

$$p_m \le p_m^{\max} \tag{13f}$$

$$\sum_{m \in \mathcal{M}^{\mathrm{O}}} f_m^{\mathrm{O}} \le f^{\mathrm{O}}. \tag{13g}$$

where Eq. (13b) is the end-to-end delay bound constraint, d_m^{Th} is the maximum tolerable delay of the task from IoT device m, Eqs. (13c) and (13d) indicate that any task is either local computing or computation offloading to the edge server, (13e) is the beamforming constraint, Eq. (13f) is the transmission power constraint, Eq. (13g) is the computation resource constraint of the edge server.

3.5 Problem decomposition

Since the JCORA problem described in Eq. (13a) involves the offloading decisions in multiple IoT devices, phase shift optimization in RIS, and computation resource allocation in edge server, which is indeed a non-convex NP-hard problem. In order to find out optimum solutions with low algorithm complexity, our approach in this paper is to decompose it into two concatenated subproblems, referred to as the CO subproblem in the involved IoT devices, and the PORA subproblem for the offloaded tasks. We use an iterative policy to update the offloading decision $\mathcal{I}$ of the CO subproblem based on the up-to-date result of the PORA subproblem. The solution of the PORA subproblem is all affected by the offloading decision of the CO subproblem in each iteration. The optimum one-shot solution of the JCORA problem is yielded when no decision update could further improve the performance of the CO and PORA subproblems. The detail of the decomposition process is described as the following.

Firstly, given the most update $(\Lambda^*, \mathtt{p}^*, \mathtt{f}^*)$ of the PORA subproblem, the current value of the energy consumption

as well as end-to-end delay of the offloaded tasks is also determined. The IoT devices can also estimate the energy consumption as well as local computation delay of the local computing tasks. Thus, as to IoT devices, the JCORA problem is reduced to the CO subproblem as follows.

$$CO: \min_{\mathcal{I}^*} E \tag{14a}$$

$$\text{s.t.} \quad \text{(13b) (13c), (13d)} \tag{14b}$$

$$\Lambda = \Lambda^*, \mathtt{p} = \mathtt{p}^*, \mathtt{f} = \mathtt{f}^*, \tag{14c}$$

where Eq. (14a) follows (13a) given Λ^*, $\mathtt{p}^*$ and $\mathtt{f}^*$; (14c) is the most updated solution of the PORA subproblem.

Secondly, the objective of the PORA subproblem is further to optimize the transmission energy consumption constrained to the delay requirements of offloaded tasks with respect to the solution of CO subproblem. That is, the PORA subproblem is formulated as,

$$\text{PORA}: \min_{\Lambda^*, \mathtt{p}^*, \mathtt{f}^*} E \tag{15a}$$

$$\text{s.t.} \quad \text{(13b), (13e), (13f), (13g)} \tag{15b}$$

$$\mathcal{I} = \mathcal{I}^*. \tag{15c}$$

In the next section, heuristic algorithms to solve both the CO and PORA subproblems are designed. The one-shot solution to the original JCORA problem will be obtained by combining the computation offloading, computation resource allocation, the RIS phase shift and transmission power optimization together via the proposed RIS-assisted DAEM scheme.

4 Proposed RIS-assisted DAEM scheme for JCORA

minimization offloading (DAECO) algorithm for solving the CO subproblem, 2) the delay-constrained transmission-energy-minimization resource optimization (DCEM) algorithm for solving the PORA subproblem. By iteratively updating the mutually affected decisions of DAECO and DCEM in limited rounds, the solution of the JCORA problem is yielded.

A heuristic RIS-assisted DAEM scheme to solve the JCORA problem is designed. As illustrated in *Algorithm* 1, the DAEM scheme consists of a) the DAECO algorithm for solving the CO subproblem, 2) the DCEM algorithm for solving the PORA subproblem. By iteratively updating the mutually affected decisions of DAECO and DCEM in limited rounds, the solution of the JCORA problem is yielded.

Algorithm 1: RIS-assisted DAEM for JCORA

Input: M, N, h^{I}, h^{D}, g^{H}, $p^{\max}$, B, σ_0, f^{O}, f^{L}, S, W, d^{Th}.

Output: E, $\mathcal{D}$, $\mathcal{I}$, $\mathcal{M}^{\mathrm{O}}$, $\mathcal{M}^{\mathrm{L}}$, Λ, $\mathtt{p}$, $\mathtt{f}$.

Initiate: Assuming all tasks would like to computation offloading, that is, let offloading decisons $I_m = 1$ for all $m \in \mathcal{M}$, then, the values of the initial edge and local computing sets are $\mathcal{M}^{\mathrm{O}} = \mathcal{M}$, $\mathcal{M}^{\mathrm{L}} = \{\}$, respectively, and the intial energy consumption $E^{\mathrm{pre}} = \infty$.

repeat

- Initiate *Algorithm* 2 to solve the CO subproblem described in (14) for obtaining up-to-date I, $\mathcal{M}^{\mathrm{O}}$ and $\mathcal{M}^{\mathrm{L}}$.
- Calculate the local computation delay D_m^{L} and corresponding energy consumption E_m^{L} for all $m \in \mathcal{M}^{\mathrm{L}}$ with (8) and (9), respectively.
- Initiate *Algorithm* 3 to solve the PORA subproblem described in (15) for obtaining the up-to-date optimum phase shift Λ, θ of the passive reflecting elements in RIS, transmission power $\mathtt{p}$ of IoT devices for task offloading, computation resource allocated $\mathtt{f}$ for $m \in \mathcal{M}^{\mathrm{O}}$, and the end-to-end delay D_m^{O} as well as the energy consumption E_m^{O} for $m \in \mathcal{M}^{\mathrm{O}}$.
- Calculate the energy consumption E with the up-to-date offloading decision $\mathcal{I}$, phase shift beamforming Λ, transmission power $\mathtt{p}$ and allocated computation resource $\mathtt{f}$.
- **if** $E < E^{pre}$ **then**
  - Let $E^{\mathrm{pre}} = E$.
- **end**

until $E \geq E^{pre}$, *or, reaches the repeat count*;

In special, as illustrated in *Algorithm* 1, at the beginning, we assume that all IoT devices are willing to offload their tasks to the edge server for computing. Then, we initiate the DAECO algorithm (see *Algorithm* 2) to update the offloading decisions of some IoT devices considering the present system state as well as the end-to-end delay of tasks and energy consumption of IoT devices. Next, we initiate the DCEM algorithm (see *Algorithm* 3) to optimize the phase shift of the RIS and computation resource allocation at edge server for reducing the delay-constrained energy consumption of offloaded tasks. After that, we again calculate the end-to-end delay and energy consumption of all tasks. We compare the energy consumption of the pre- and after- DAECO and DCEM policy-interchanging-update. If the policy update reduces the energy consumption, then we repeat the DAECO and DCEM policy-interchanging-update process; otherwise, or, the iteration reaches the repeat count, we stop the policy-interchanging update, and yield the solution of the JCORA problem by joining the up-to-date solutions given by DAECO for the CO subproblem and DCEM for the PORA subproblem.

4.1 DAECO for CO

Given the computation resource allocation policy of the edge server for offloaded tasks and the phase shift beamforming of the RIS as well as the transmission power, the end-to-end delay and energy consumption of the offloading tasks are determined. For the tasks determined to local computing, since per IoT device handles one local task in parallel, the end-to-end delay as well as energy consumption is also estimable. However, since the offloading decision switching of one IoT device would affect the end-to-end delay and energy consumption performance of other IoT devices due to the computation resource sharing at the edge server and phase shift beamforming optimization for all offloading tasks, it is better to iteratively update the offloading decisions of some IoT devices for reducing the energy consumption constrained to delay requirement. Based on the above observation, we design an iterative delay-aware energy consumption minimization offloading algorithm as illustrated in *Algorithm* 2 for solving the CO subproblem.

Algorithm 2: DAECO for CO subproblem

Input: $\mathcal{M}$,$\mathcal{M}^{\mathrm{O}}$, $\mathcal{M}^{\mathrm{L}}$, $\mathcal{I}$, N, h^{I}, h^{D}, g^{H}, $p^{\max}$, B, σ_0, f^{O}, f^{L}, S, W, d^{Th}.

Output: $\mathcal{I}^*$, $\mathcal{M}^{\mathrm{O}*}$, $\mathcal{M}^{\mathrm{L}*}$.

Initiate: Let $\mathcal{M}^{\mathrm{O}''} = \{\}$ represent the determined offloading IoT device set.

repeat

  Find m' who would experience the longest end-to-end delay from $\mathcal{M}^{\mathrm{O}}$. That is,

$$m' = \mathrm{argmax}_{m \in \mathcal{M}^{\mathrm{O}}} D^{\mathrm{O}}_m. \tag{16}$$

  Estimate the local computation delay $D^{\mathrm{L}}_{m'}$ and corresponding energy consumption $E^{\mathrm{L}}_{m'}$ of m' with (8) and (9), respectively.

  if $max(D^{O}_{m'}, D^{L}_{m'}) \leq d^{Th}_{m'}$, **then**

    if $E^{L}_{m'} < E^{O}_{m'}$ **then**

      Switch the offloading decision of m'. That is, let $I_{m'} = 0$, $\mathcal{M}^{\mathrm{O}} = \mathcal{M}^{\mathrm{O}} \setminus \{m'\}$, $\mathcal{M}^{\mathrm{L}} = \mathcal{M}^{\mathrm{L}} \cup \{m'\}$.

    else

      Keep the present offloading decision of m'. That is, move m' from $\mathcal{M}^{\mathrm{O}}$ to $\mathcal{M}^{\mathrm{O}''}$ by letting $\mathcal{M}^{\mathrm{O}} = \mathcal{M}^{\mathrm{O}} \setminus \{m'\}$ and $\mathcal{M}^{\mathrm{O}''} = \mathcal{M}^{\mathrm{O}''} \cup \{m'\}$.

    end

  else if $D^{O}_{m'} > d^{Th}_{m}$, **then**

    if $D^{L}_{m'} \leq d^{Th}_{m} \mid E^{L}_{m'} < E^{O}_{m'}$ **then**

      Switch the offloading decision of m'. That is, let $I_{m'} = 0$, $\mathcal{M}^{\mathrm{O}} = \mathcal{M}^{\mathrm{O}} \setminus \{m'\}$, $\mathcal{M}^{\mathrm{L}} = \mathcal{M}^{\mathrm{L}} \cup \{m'\}$.

    end

  else

    Keep the present offloading decision of m'. That is, move m' from $\mathcal{M}^{\mathrm{O}}$ to $\mathcal{M}^{\mathrm{O}''}$ by letting $\mathcal{M}^{\mathrm{O}} = \mathcal{M}^{\mathrm{O}} \setminus \{m'\}$ and $\mathcal{M}^{\mathrm{O}''} = \mathcal{M}^{\mathrm{O}''} \cup \{m'\}$.

  end

until $\mathcal{M}^{O} = \{\}$;

Finally: Let $\mathcal{I}^* = \mathcal{I}$, $\mathcal{M}^{\mathrm{O}*} = \mathcal{M}^{\mathrm{O}''}$, $\mathcal{M}^{\mathrm{L}*} = \mathcal{M}^{\mathrm{L}}$.

As illustrated in *Algorithm* 2, in order to guarantee the end-to-end delay of as more as possible offloaded tasks, in each round of iteration, we choose an offloaded IoT device m' who would experience the longest end-to-end delay, which generally has the worst link quality among offloaded IoT devices, even under the assistance of RIS. Then, we estimate its local computation delay and corresponding energy consumption. If its offloading decision switching could reduce the energy consumption constrained to delay requirement or satisfying the end-to-end delay requirement of the task, then we update its offloading decision (see lines 7 and 13 of *Algorithm* 2); otherwise, the offloading decision is reserved (see lines 9 and 16 of *Algorithm* 2). We repeat the offloading decision switching steps until all offloaded IoT devices have been ergodic. After the ergodic process, we return the up-to-date offloading decision $\mathcal{I}^*$ as well as local and offloaded IoT device sets $\mathcal{M}^{\mathrm{L}*}$ and $\mathcal{M}^{\mathrm{O}*}$ back to *Algorithm* 1 for assisting in solving the PORA subproblem and yielding the one-shot solution.

4.2 DCEM for PORA

Given the offloading decision $\mathcal{I}^*$, the number of offloaded IoT devices $M^{\mathrm{O}*}$ is also determined, then the PORA subpproblem is equivalent to the joint phase shift beamforming optimization, transmission power reduction and computation resource allocation for these offloaded tasks/IoT devices. We design a DCEM algorithm to solve it.

As illustrated in *Algorithm* 3, the DCEM algorithm consists of three subprocesses, computation resource allcation, iterative beamforming optimization and transmission power reduction.

Algorithm 3: DCEM for PORA subproblem

Input: $\mathcal{M}^{\mathrm{O}}$, $\mathcal{I}$, N, h^{I}, h^{D}, g^{H}, $p^{\max}$, B, σ_0, f^{O}, S, W, d^{Th}.

Output: $f_m^{\mathrm{O}*}$, p_m^*, D_m^{O}, E_m^{O} for all $m \in \mathcal{M}^{\mathrm{O}*}$, Λ^*, θ^*.

1 **Initiate**: Let phase shift $\theta_n = 0$ for $n \in \mathcal{N}$, transmission power $p_m = p^{\max}$ for all $m \in \mathcal{M}^{\mathrm{O}}$.

2 **Computation resource allocation:**

3 For a task with size W_m for $m \in \mathcal{M}^{\mathrm{O}}$, its resource weight is derived by

$$\alpha_m = \frac{W_m}{\sum_{m \in \mathcal{M}^{\mathrm{O}}} W_m}. \tag{17}$$

Then, the computing rate of the offloaded task m is derived by

$$f_m^{\mathrm{O}*} = \alpha_m f^{\mathrm{O}}. \tag{18}$$

4 Calculate the computation delay $D_m^{\mathrm{O,\,CPU}}$ for $m \in \mathcal{M}^{\mathrm{O}}$ at the edge server using (4).

5 **Iterative beamforming optimization:**

6 A sum-SNR-maximization beamforming optimization method (see *Algorithm* 4) is designed to obtain Λ^* and θ^*.

7 Calculate the transmission rate R_m for $m \in \mathcal{M}^{\mathrm{O}}$ using (3).

8 Estimate $D_m^{\mathrm{O,\,Tx}}$ for $m \in \mathcal{M}^{\mathrm{O}}$ using (2).

9 **Transmission power reduction:**

10 Choose p_m^* for $m \in \mathcal{M}^{\mathrm{O}}$ as the solution to the following:

$$\min_{p_m^*} \quad E_m^{\mathrm{O,\,Tx}} \tag{19a}$$

$$\text{s.t.} \quad D_m^{\mathrm{O}} \le d_m^{\mathrm{Th}}, \tag{19b}$$

$$0 < p_m \le p_m^{\max}, \tag{19c}$$

$$\Lambda = \Lambda^*, f_m^{\mathrm{O}} = f_m^{\mathrm{O}*}, \tag{19d}$$

$$(1), (7)\ . \tag{19e}$$

Computation resource allocation: The computation resource of the edge server for the competing offloaded tasks is allocated based on the weighted fair policy. That is, the allocated computation resource of an offloaded task is proportional to its computing amount. The motivation of using the weighted fair policy is that, it provies a workload based fairness to all competing tasks. In addition, it is easy to implement with low time-complexity.

Iterative beamforming optimization: As shown in Eqs. (6) and (7), in the offloading case, the transmission power of the IoT devices and the upload delay mainly affect the energy consumption of IoT devices. Furthermore, as shown in Eqs. (2) and (3), optimizing the RIS phase shift Λ, the SNR could be improved, thus reducing the upload delay, leading to the energy consumption reduction. Accordingly, given the computation delay $D_m^{\mathrm{O,\,CPU}}$ $\forall m \in \mathcal{M}^{\mathrm{O}}$ at edge server and the transmission power p_m^* $\forall m \in \mathcal{M}^{\mathrm{O}}$, the energy consumption minimization problem for offloaded tasks/IoT devices is equivalent to the SNR maximization by reconfiguring the RIS phase shift problem (called BeamOpt), which is formulated as

$$\text{BeamOpt}: \max_{\Lambda^*} \Psi = \sum_{m\in\mathcal{M}^{\mathrm{O}}} \Psi_m \text{s.t.} \quad \Psi_m = |g^H \Lambda h_m^I + h_m^{\mathrm{D}}|^2 \tag{20a}$$

$$\Lambda = \mathrm{diag}(e^{j\theta_1}, e^{j\theta_2}, \ldots, e^{j\theta_N}) \tag{20b}$$

$$0 \le \theta_n \le 2\pi, n \in \mathcal{N} \tag{20c}$$

$$p_m = p_m^*. \tag{20d}$$

where Eq. (20a) follows Eq. (3), (20c) follows Eq. (13e).

Since

$$|g^H \Lambda h_m^{\mathrm{I}} + h_m^{\mathrm{D}}| \le |g^H \Lambda h_m^I| + |h_m^{\mathrm{D}}|, \tag{21}$$

the equality holds if and only if the phase of $g^H \Lambda h_m^{\mathrm{I}}$ equals to that of h_m^{D} [6]. Let $\arg(h_m^{\mathrm{D}}) \triangleq \phi_{m,0}$, then, for any offloading IoT device m, to maximize Ψ_m is equivalent to maximize $|g^H \Lambda h_m^{\mathrm{I}}|^2$ constrained to $\arg(g^H \Lambda h_m^{\mathrm{I}}) = \phi_{m,0}$.

Let $\mathrm{v} = (e^{j\theta_1}, e^{j\theta_2}, \ldots, e^{j\theta_N})^H$, then we have $g^{\mathrm{H}} \Lambda h_m^{\mathrm{I}} = \mathrm{v}^H \mathrm{diag}(g^{\mathrm{H}}) h_m^{\mathrm{I}}$. Accordingly, with Eq. (21), the problem described in Eq. (20) is equivalent to

$$\max_{\mathrm{v}} \sum_{m\in\mathcal{M}^{\mathrm{O}}} |\mathrm{v}^H \mathrm{diag}(g^{\mathrm{H}}) h_m^{\mathrm{I}}|^2 \tag{22a}$$

$$\text{s.t} \quad |v_n| = 1, \forall n = 1, 2, \ldots, N \tag{22b}$$

$$\arg(\mathrm{v}^H \mathrm{diag}(g^{\mathrm{H}}) h_m^{\mathrm{I}}) = \phi_{m,0}, \forall m \in \mathcal{M}^{\mathrm{O}}. \tag{22c}$$

For $m \in \mathcal{M}^{\mathrm{O}}$, the optimum $|\mathrm{v}^H \mathrm{diag}(g^{\mathrm{H}}) h_m^{\mathrm{I}}|^2$ is yielded by $\mathrm{v}_m^* = e^{j(\phi_{m,0} - \arg(\mathrm{diag}(g^{\mathrm{H}}) h_m^{\mathrm{I}}))}$. That is, the nth phase shift of RIS considering the user-specific channel quality could be given by

$$\begin{aligned} \theta_{m,n} &= \phi_{m,0} - \arg(g_n^{\mathrm{H}} h_{n,m}^{\mathrm{I}}) \\ &= \phi_{m,0} - \arg(g_n^{\mathrm{H}}) - \arg(h_{n,m}^{\mathrm{I}}), \end{aligned} \tag{23}$$

where g_n^{H} is the nth element of g^{H}, $h_{n,m}^{\mathrm{I}}$ is the effective channel of the nth element of the RIS to IoT device m.

Since the angle of incidence from distinct IoT devices to the RIS may be different, as described in Eq. (23), the optimum θ for distinct IoT devices might be different. In this case, we use a sum-SNR-maximization beamforming optimization algorithm as illustrated in *Algorithm* 4 to iteratively search an optimum v^* among v_m^* obtaining via Eq. (23) for an optimum Eq. (22a). The detail is illustrated in *Algorithm* 4.

Algorithm 4: Sum-SNR-maximization iterative beamforming optimization

Input: $\mathcal{M}^{\mathrm{O}}$, N, h^{I}, h^{D}, g^{H}, p_m for $m \in \mathcal{M}^{\mathrm{O}}$, B, σ_0.

Output: Λ^*, θ^*.

Initiate: $\omega_m = \frac{(h_m^{\mathrm{D}})^H}{\|h_m^{\mathrm{D}}\|}$ for $m \in \mathcal{M}^{\mathrm{O}}$, $\mathrm{SNR}^{\mathrm{tot}} = 0$.

repeat

  for $m \in \mathcal{M}^{O}$ **do**

    Calculate $\theta 0_m$ for $m \in \mathcal{M}^{\mathrm{O}}$ by

$$\theta 0_m = \arg\left((h_m^{\mathrm{D}})^H \omega_m\right), \quad (24)$$

    where $\arg X$ indicates the phase shift of X.

    The phase shift of the nth passive element for $n \in \mathcal{N}$ is calculated by

$$\begin{cases} \theta 1_{m,n} = \arg(g_n), \\ \theta 2_{m,n} = \arg(h_{n,m}^{\mathrm{I}} \omega_m), \\ \theta_{m,n} = \theta 0_m - \theta 1_{m,n} - \theta 2_{m,n}. \end{cases} \quad (25)$$

    Calculate the user-specific reflection coefficient matrix using (26).

$$\Lambda_m = \mathrm{diag}(e^{j\theta_{m,1}}, \ldots, e^{j\theta_{m,N}}). \quad (26)$$

    Evaluate the sum SNR with Λ_m by

$$\mathrm{SNR}_m^{\mathrm{tot}} = |g\Lambda_m (h_m^{\mathrm{I}})^H + (h_m^{\mathrm{D}})^H|^2 + \sum_{m' \in \mathcal{M}^{\mathrm{O}}, m' \neq m} |g\Lambda_m (h_{m'}^{\mathrm{I}})^H + (h_{m'}^{\mathrm{D}})^H|^2. \quad (27)$$

    Choose $\Lambda_{m'}$ from $\{\Lambda_m : \forall m \in \mathcal{M}^{\mathrm{O}}\}$, which satisfies

$$\Lambda_{m'} = \mathrm{argmax}_{\Lambda_m} \mathrm{SNR}_m^{\mathrm{tot}}, \quad (28)$$

    then set $\Lambda = \Lambda_{m'}$, $\theta = \theta_{m'}$.

    Update ω_m using (29)

$$\omega_m = \frac{(g^{\mathrm{B}} \Lambda (h_m^{\mathrm{I}})^H + (h_m^{\mathrm{D}})^H)^H}{\|g^{\mathrm{B}} \Lambda (h_m^{\mathrm{I}})^H + (h_m^{\mathrm{D}})^H\|}, \forall m \in \mathcal{M}^{\mathrm{O}}. \quad (29)$$

  end

until *reaches the repeat count or no update happens*;

Transmission power reduction: Given $f_m^{\mathrm{O}*}$, the computation delay of offloaded tasks is determined, thus, the tolerable data upload delay constrained to end-to-end delay could be estimated. Furthermore, as shown in Eq. (7) and Eq. (3), given the maximum tolerable upload delay $D_m^{\mathrm{O,Tx}}$, the transmission power could be optimized for reducing the energy consumption. Accordingly, we use the method described in *transmission power reduction* of *Algorithm* 3 to derive the optimum transmission power for reducing the transmission energy consumption.

5 Performance evaluation

Simulation results are provided in this section to demonstrate the effectiveness of the proposed RIS-assisted DAEM for energy efficiency constrained to delay requirements.

5.1 Parameter setting

In simulations, the number of IoT devices and passive reflecting elements in RIS are set to $M = 20$ and $N = 20$, respectively. The positions of the base station and RIS are set to (0, 0, 0)m and (20, 0, 0)m, respectively. The IoT devices are randomly distributed in a circular zone, where the minimum and maximum horizontal (x-axis) distances from the base station are $d_{\min} = 50$m and $d_{\max} = 55$m respectively, the radius of the circular zone is set to $r = (d_{\max} - d_{\min})/2$, while the vertical (z-axis) distance is set to $d_v = 3$m from the base station, as illustrated in Fig. 2. The large-scale fading model is given by $L(d) = C_0(d)^{-a}$ [32], where $C_0 = -30$dB, α for IoT devices to BS, IoT devices to RIS, and RIS to BS, are set as 5, 2.8, 2 respectively.

The computation capabilities of the edge server and IoT device are set to $f^{\mathrm{O}} = 20$ GHz and $f_m^{\mathrm{L}} = 0.6$GHz for $m \in \mathcal{M}$, respectively. The task size is randomly distributed between 0.5Mb and 0.8Mb in bits, corresponding to 2.95G CPU cycles and 4.72G CPU cycles, respectively. The delay bound of a task is set to $d_m^{\mathrm{Th}} = 10$ms for $m \in \mathcal{M}$. The other parameters are set as follows: $\sigma_0 = 10^{-11}$, $\alpha = 1.0$, $\beta = 0.25 \times 10^{-18}$.

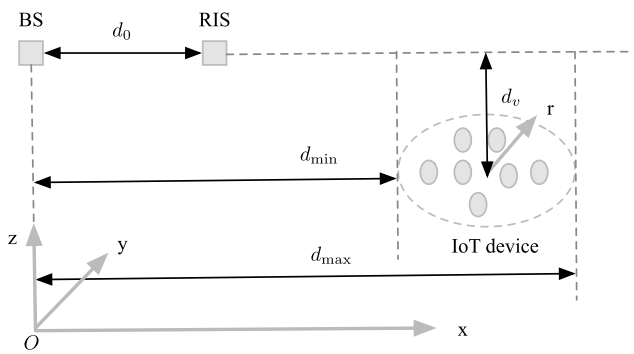


Fig. 2 Simulation setup (top view)

This paper evaluates the efficiency of the proposal with the following four benchmarked schemes.

- *Local computing Greedy (LG)*: All tasks are computed at local source nodes [32].
- *RIS-assisted Edge computing Greedy (RIS-assisted EG)*: All tasks are offloaded to the edge server via the RIS-assisted wireless network for edge computing. In special, the offloading decisions are made according to the edge computing greedy policy [36]. The optimum phase shift of the RIS is solved via semi-definite relaxation (SDR) method as described in [6, 30], while the optimum transmission power is determined with the delay-aware energy minimization transmission policy as shown in lines 9-10 of *Algorithm* 3.
- *Delay-Constrained Computation Offloading without RIS (DCCO)*: The offloading decisions are made according to the delay requirements of tasks. That is, an offloading policy satisfying the delay bound of a task is chosen [36]. For the task determined to local computing, similar method as described in line 4 of *Algorithm* 1 is used to allocate CPU resource of local computing nodes to local computing tasks. For the tasks determined to edge computing, delay-constrained energy minimization transmission policy (similar to *Algorithm* 3) is adopted.
- *Energy-efficient Task Offloading Strategy (ETOS)*: ETOS is a hybrid method based on particle swarm optimization (PSO) and grey wolf optimizer (GWO) to solve the energy-efficient task offloading problem, which considers efficient resource allocation such as sub-carries, power, and bandwidth for offloading to guarantee minimum energy consumption along with satisfying delay requirements [37].

To evaluate the delay guarantee capability of the investigated schemes, we introduce a delay guarantee ratio defined as follows.

$$G = \frac{\text{The number of delay guaranteed tasks}}{\text{Total tasks}} \times 100\%. \quad (24)$$

The higher of G, the higher delay guarantee capability of the investigated scheme.

5.2 Adaptive to task size

First, we investigate the impact of the task size on the energy efficiency constrained to delay requirement. As shown in Fig. 3, the energy consumptions of all the investigated schemes increase with the increasing task size, which is accordance with our intuition. The DCCO, ETOS and RIS-assisted DAEM schemes outperform the LG and RIS-assisted EG schemes by given lower energy consumption under various task sizes, which demonstrates that the

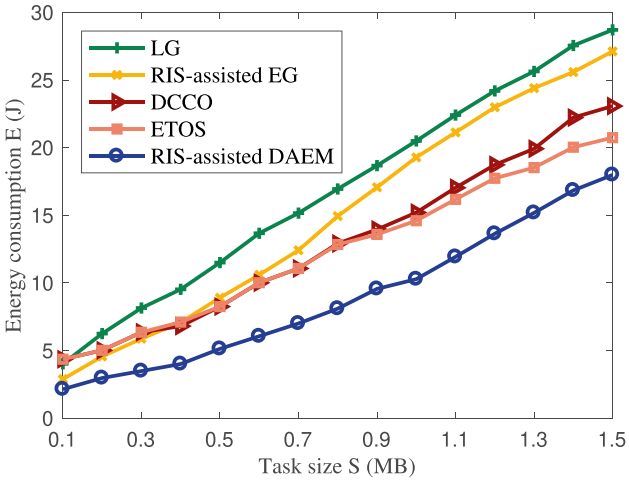


Fig. 3 Energy consumption with respect to task size

device-edge collaborative computation offloading schemes, which consider the delay requirements and energy-efficiency, outperform both of the device-single and the edge-single computation offloading schemes.

RIS-assisted DAEM outperforms the other investigated schemes by always consuming the lowest energy and providing the highest delay guarantee ratio under various task sizes, as illustrated in Figs. 3 and 4. This is because, under RIS-assisted DAEM, the IoT device would adaptively select a computing node (e.g., local node or edge server) for energy reduction constrained to the delay requirement of the corresponding task. In addition, energy efficiency resource allocation algorithms, such as delay-constrained CPU rate allocation and delay-aware transmission power determination, as well as phase shift beamforming optimization are collaboratively used in RIS-assisted DAEM to further improve the energy efficiency.

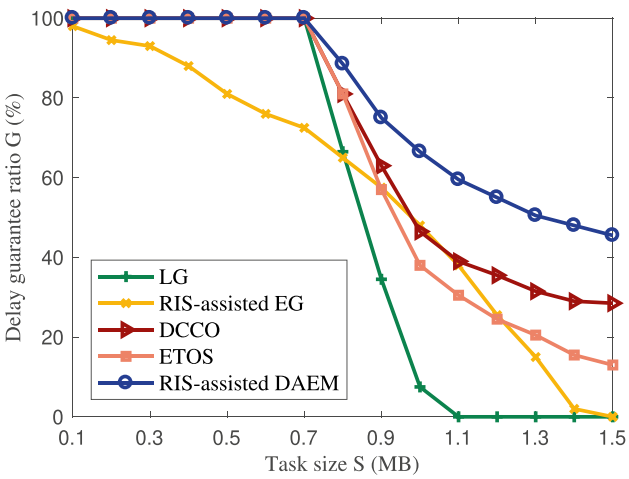


Fig. 4 Delay guarantee ratio with respect to task size

LG provides the worst performance by given the highest energy consumption (see Fig. 3). In addition, when the task size exceeds 0.7MB/task (equivalently, the CPU computing amount exceeds 4.13 G CPU cycles/task), the delay guarantee ratio given by LG decreases quickly. When the task size reaches 1.1 MB/task (equivalently, the CPU computing amount reaches 6.49 G CPU cycles/task), the delay guarantee ratio given by LG reduces to zero, as illustrated in Fig. 4. This is because, without the computation assistance of edge server, the limited computation resource of an end device is difficult to support the low-delay and high performance computation tasks.

Notice that, when the average task size reaches 0.7MB/task (equivalently, the CPU computing amount reaches 4.13 G CPU cycles/task), the average workload of the system has been 2.6 times of the MEC system's computing capacity. Therefore, it is unsurprising that, when the average task size exceeds 0.7MB/task, the delay guarantee ratio given by all the investigated schemes decrease with the increasing task size. However, RIS-assisted DAEM outperforms the other investigated schemes by providing the highest delay guarantee ratio under various task sizes. For example, as illustrated in Fig. 4, when the task size reaches 1.5 MB/task, the average workload of the system has been 5.53 times of the MEC system's computing capacity (severe heavy load state), the delay guarantee ratio given by RIS-assisted DAEM could still be greater than 40%, while those given by LG and RIS-assisted EG has reached zero.

When the workload is slight (e.g., the task size is less than 0.5 MB/task), RIS-assisted EG consumes similar energy with DCCO and ETOS. However, when the task size continually increases, the energy consumption of RIS-assisted EG increases quickly and exceeds that of DCCO and ETOS, as shown in Fig. 3. This is because, although the assistance of RIS could reduce the transmission energy consumption, however, under the full offloading scheme (e.g., RIS-assisted EG), the transmission energy consumption is still larger than those of the partial offloading schemes (e.g., DCCO, ETOS and RIS-assisted DAEM).

5.3 Adaptive to user number

We further observe the performance by varying the number of IoT devices. We set the delay bound to $d_m^{\mathrm{Th}} = 6$ms for $m \in \mathcal{M}$. Other parameters are the same as described in Section 5.1.

The energy consumption and delay guarantee ratio of the investigated schemes are shown in Figs. 5 and 6, respectively. Since the number of tasks increases with the increasing number of IoT devices, it is unsurprising that the energy consumption given by all the investigated schemes increases with the increasing number of IoT devices.

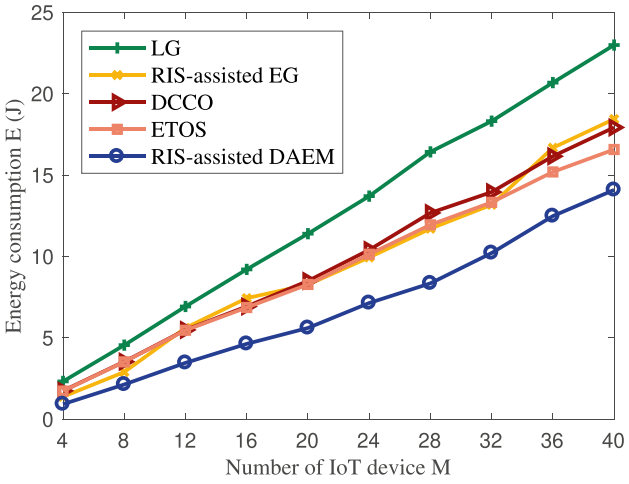


Fig. 5 Energy consumption with respect to IoT user number

The effectiveness of RIS-assisted DAEM is again validated by given the lowest energy consumption (see Fig. 5) while providing the highest delay guarantee ratio (see Fig. 6) under various IoT device numbers. This is because, RIS-assisted DAEM can dynamically switch between local computing and edge server computing adaptive to the varying of IoT device numbers to reduce the energy consumption considering the tolerable delay of tasks.

LG again consumes the highest energy for processing similar tasks in comparison with other investigated schemes under various number of IoT devices, as illustrated in Fig. 5. However, the delay guarantee ratio given by LG is stable under various number of IoT devices, as shown in Fig. 6. This is because, the per user's local processing task under LG does not change with the varying of IoT user numbers. Thus, it is unsurprising that the delay guarantee ratio given by LG would not change with the varying of IoT user numbers. The delay guarantee ratio given by RIS-assisted EG

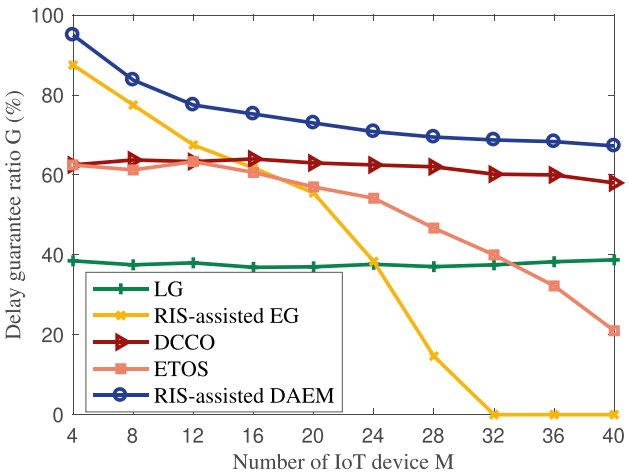


Fig. 6 Delay guarantee ratio with respect to IoT user number

decreases quickly with the increasing number of users, as illustrated in Fig. 6. The reason is that, with the increasing number of users, the offloading workload given by the full offloading scheme (e.g., EG and its variants) increases the fastest in comparison with partial offloading schemes (e.g., DCCO, ETOS and RIS-assisted DAEM). The delay guarantee ratio given by DCCO is better than that by ETOS, since DCCO takes delay guarantee as its optimum object, while ETOS takes energy-efficiency as its optimum object. Thus, the energy consumption of ETOS is less than that of DCCO, as shown in Fig. 5.

5.4 Adaptive to delay bound

Finally, we investigate the impact of delay bound on the performance. We can see in Fig. 7 that, the energy consumption of RIS-assisted EG, DCCO and RIS-assisted DAEM decrease with the increasing delay bound. The reason is that, DCCO and RIS-assisted DAEM are delay-aware computation offloading schemes, they can dynamically switch between local computing and edge server computing for delay guarantee. Furthermore, the allocated CPU rate as well as the transmission power could be reduced with the loosing delay bound. Therefore, the energy consumption given by both DCCO and RIS-assisted DAEM decrease explicitly with the increasing delay bound. Under RIS-assisted EG, the allocated CPU rate and the transmission power could also be reduced with the loosing delay bound, thus the energy consumption of RIS-assisted EG also decreases with the increasing delay bound.

Since LG is delay-unaware scheme, the energy consumption of LG does not change with the varying of delay bound, as shown in Fig. 7. Since under ETOS, the varying of delay bound in a small range (e.g., a low delay bound in a range of 1ms to 10ms) has less impact on the offloading decisions in comparison with the energy consumption, the energy consumption (equivalently, offloading decision) of ETOS does not change with the varying of delay bound. Indeed, as shown in Fig. 7, the varying of energy consumption given by ETOS is due to the varying of wireless transmission environment.

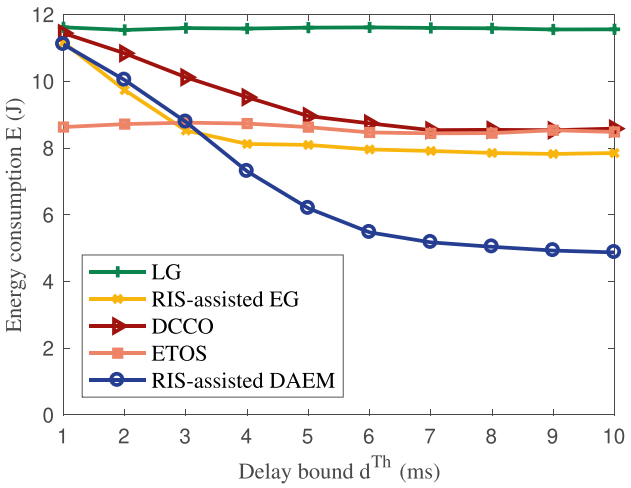


Fig. 7 Energy consumption with respect to delay bound

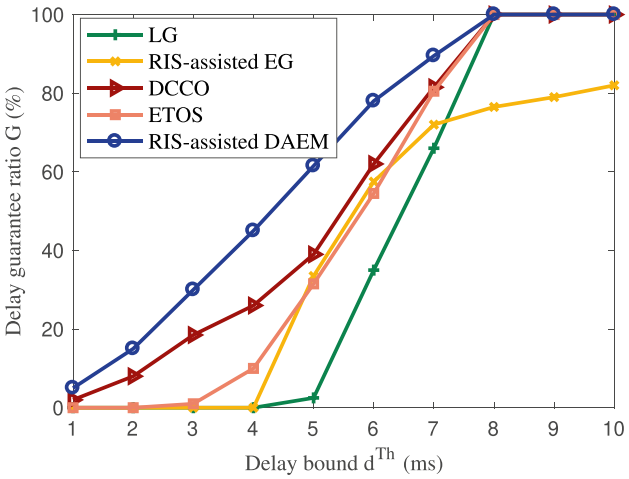


Fig. 8 Delay guarantee ratio with respect to delay bound

Compared to the workload (task request), the CPU resource of both IoT device and edge server is not sufficient for extremely low delay (e.g., $<$ 8ms) provisioning in the simulation setting, furthermore, the distance between data source (IoT device) and base station is a little far ($>$ 50m), it is unsurprising that all the investigated schemes could not bound the delays of all tasks when the delay bound is less than 8ms, as illustrated in Fig. 8. However, the delay-aware offloading schemes, such as DCCO and RIS-assisted DAEM, still outperforms the delay-unaware offloading schemes, such as LG, RIS-assisted EG and ETOS. Indeed, the increasing of delay guarantee ratio given by LG and ETOS is mostly due to the statistical property.

The effectiveness of RIS-assisted DAEM is again validated by that, the energy consumption given by RIS-assisted DAEM reduces faster and is lower than those of the other investigated schemes when $d^{\text{Th}} \geq 3$ms, in addition, the delay guarantee ratio given by RIS-assisted DAEM increases faster and is higher than those of the other investigated schemes, as illustrated in Figs. 7 and 8, respectively.

6 Conclusion

This paper has proposed a RIS-assisted device-edge collaborative edge computing scheme, termed RIS-assisted DAEM, for addressing the problem of energy consumption minimization constrained to the delay requirements in IIoT environments. In the proposal, a RIS is deployed between IoT devices and edge server for improving the wireless

performance. To find out optimum offloading decisions, e.g., local computing or edge computing, a DAECO algorithm is proposed to address the CO subproblem. To find out optimum phase shift of RIS, computation resource allocation strategy at edge server and transmission power at IoT devices, for offloading tasks, the DCEM algorithm is proposed to solve the PORA subproblem. By joining the up-to-date solutions of CO and PORA via iterative RIS-assisted DAEM, the goal of energy consumption minimization constrained to the delay requirements is yielded. Simulation results show that the proposed scheme can significantly reduce the energy consumption of IoT devices constrained to the delay requirements. RIS-assisted DAEM also outperforms the benchmarked schemes in terms of delay guarantee.

Author contributions Mian Guo and Mithun Mukherjee wrote the main part of the manuscript. Mian Guo developed the model and performed experiments. Chengyuan Xu performed the experiments. All authors read and approved the final manuscript.

Funding This work was supported in part by the National Natural Science Foundation of China under Grant 62273109 and 61901128, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (21KJB510032).

Data availability Not applicable.

Declarations

Ethics approval This work does not contain any studies with human participants or animals performed by any of the authors.

Consent to participate Not applicable.

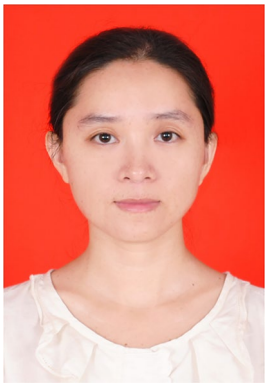
Consent for publication Not applicable.

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Qiu T, Chi J, Zhou X, Ning Z, Atiquzzaman M, Wu DO (2020) Edge computing in industrial internet of things: Architecture, advances and challenges. IEEE Commun Surv Tutor 22(4):2462–2488. https://doi.org/10.1109/COMST.2020.3009103
2. Xiang Z, Zheng Y, He M, Shi L, Wang D, Deng S, Zheng Z (2022) Energy-effective artificial internet-of-things application deployment in edge-cloud systems. Peer-to-Peer Networking and Applications 15(2):1029–1044. https://doi.org/10.1007/s12083-021-01273-5
3. Zhang J, Liu J, Ma S, Wen CK, Jin S (2021) Large system achievable rate analysis of ris-assisted MIMO wireless communication with statistical CSIT. IEEE Trans Wirel Commun 20(9):5572–5585. https://doi.org/10.1109/TWC.2021.3068494
4. Zuo J, Liu Y, Ding Z, Song L, VincentPoor H (2022) Joint design for simultaneously transmitting and reflecting (STAR) RIS assisted NOMA systems. IEEE Transactions on Wireless Communications pp 1–1, https://doi.org/10.1109/TWC.2022.3197079
5. Wei X, Shen D, Dai L (2021) Channel estimation for RIS assisted wireless communications part I: Fundamentals, solutions, and future opportunities. IEEE Commun Lett 25(5):1398–1402. https://doi.org/10.1109/LCOMM.2021.3052822
6. Wu Q, Zhang R (2019) Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming. IEEE Trans Wirel Commun 18(11):5394–5409. https://doi.org/10.1109/TWC.2019.2936025
7. Mao Y, You C, Zhang J, Huang K, Letaief KB (2017) A survey on mobile edge computing: The communication perspective. IEEE Commun Surv Tutor 19(4):2322–2358. https://doi.org/10.1109/COMST.2017.2745201
8. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: Vision and challenges. IEEE Internet Things J 3(5):637–646. https://doi.org/10.1109/JIOT.2016.2579198
9. Zhou Z, Feng J, Chang Z, Shen X (2019) Energy-efficient edge computing service provisioning for vehicular networks: A consensus ADMM approach. IEEE Trans Veh Technol 68(5):5087–5099. https://doi.org/10.1109/TVT.2019.2905432
10. Zhang K, Mao Y, Leng S, Zhao Q, Li L, Peng X, Pan L, Maharjan S, Zhang Y (2016) Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE Access 4:5896–5907. https://doi.org/10.1109/ACCESS.2016.2597169
11. Song Z, Hao Y, Liu Y, Sun X (2021) Energy-efficient multiaccess edge computing for terrestrial-satellite internet of things. IEEE Internet Things J 8(18):14202–14218. https://doi.org/10.1109/JIOT.2021.3068141
12. Zhou H, Jiang K, Liu X, Li X, Leung VCM (2022) Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing. IEEE Internet Things J 9(2):1517–1530. https://doi.org/10.1109/JIOT.2021.3091142
13. Wang Q, Guo S, Liu J, Yang Y (2019) Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing. Sustainable Computing: Informatics and Systems 21:154–164. https://doi.org/10.1016/j.suscom.2019.01.007
14. Yi C, Cai J, Su Z (2020) A multi-user mobile computation offloading and transmission scheduling mechanism for delay-sensitive applications. IEEE Trans Mob Comput 19(1):29–43. https://doi.org/10.1109/TMC.2019.2891736
15. Kuang Z, Ma Z, Li Z, Deng X (2021) Cooperative computation offloading and resource allocation for delay minimization in mobile edge computing. Journal of Systems Architecture 118. https://doi.org/10.1016/j.sysarc.2021.102167
16. Ale L, Zhang N, Fang X, Chen X, Wu S, Li L (2021) Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning. IEEE Transactions on Cognitive Communications and Networking 7(3):881–892. https://doi.org/10.1109/TCCN.2021.3066619
17. Shahryari OK, Pedram H, Khajehvand V, Dehghan TakhtFooladi M (2020) Energy-efficient and delay-guaranteed computation offloading for fog-based IoT networks. Computer Networks 182. https://doi.org/10.1016/j.comnet.2020.107511
18. Liu L, Guo X, Chang Z, Ristaniemi T (2019) Joint optimization of energy and delay for computation offloading in cloudlet-assisted mobile cloud computing. Wireless Netw 25(4):2027–2040. https://doi.org/10.1007/s11276-018-1794-0
19. Chen Z, Zheng H, Zhang J, Zheng X, Rong C (2022) Joint computation offloading and deployment optimization in multi-UAV-enabled MEC systems. Peer-to-Peer Networking and Applications 15(1):194–205. https://doi.org/10.1007/s12083-021-01245-9
20. Abeywickrama S, Zhang R, Wu Q, Yuen C (2020) Intelligent reflecting surface: Practical phase shift model and beamforming optimization. IEEE Trans Commun 68(9):5849–5863. https://doi.org/10.1109/TCOMM.2020.3001125
21. Zheng B, You C, Mei W, Zhang R (2022) A survey on channel estimation and practical passive beamforming design for intelligent

reflecting surface aided wireless communications. IEEE Commun Surv Tutor 24(2):1035–1071. https://doi.org/10.1109/COMST.2022.3155305
22. Wei X, Shen D, Dai L (2021) Channel estimation for RIS assisted wireless communications part II: An improved solution based on double-structured sparsity. IEEE Commun Lett 25(5):1403–1407. https://doi.org/10.1109/LCOMM.2021.3052787
23. Wu Q, Zhou X, Schober R (2021) IRS-assisted wireless powered NOMA: Do we really need different phase shifts in DL and UL? IEEE Wirel Commun Lett 10(7):1493–1497. https://doi.org/10.1109/LWC.2021.3072502
24. Van Chien T, Papazafeiropoulos AK, Tu LT, Chopra R, Chatzinotas S, Ottersten B (2021) Outage probability analysis of IRS-assisted systems under spatially correlated channels. IEEE Wirel Commun Lett 10(8):1815–1819. https://doi.org/10.1109/LWC.2021.3082409
25. Al-Hilo A, Samir M, Elhattab M, Assi C, Sharafeddine S (2022) RIS-assisted UAV for timely data collection in IoT networks. IEEE Syst J pp. 1–12. https://doi.org/10.1109/JSYST.2022.3215279
26. Sankar RP, Chepuri SP (2022) Beamforming in hybrid ris assisted integrated sensing and communication systems. https://doi.org/10.23919/EUSIPCO55093.2022.9909562
27. Zhang H, He X, Wu Q, Dai H (2021) Spectral graph theory based resource allocation for IRS-assisted multi-hop edge computing. https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484578
28. Mei H, Yang K, Shen J, Liu Q (2021) Joint trajectory-task-cache optimization with phase-shift design of RIS-assisted UAV for MEC. IEEE Wirel Commun Lett 10(7):1586–1590. https://doi.org/10.1109/LWC.2021.3074990
29. Bai T, Pan C, Deng Y, Elkashlan M, Nallanathan A, Hanzo L (2020) Latency minimization for intelligent reflecting surface aided mobile edge computing. IEEE J Sel Areas Commun 38(11):2666–2682. https://doi.org/10.1109/JSAC.2020.3007035
30. Sun C, Ni W, Bu Z, Wang X (2022) Energy minimization for intelligent reflecting surface-assisted mobile edge computing. IEEE Trans Wirel Commun 21(8):6329–6344. https://doi.org/10.1109/TWC.2022.3148296
31. Zq Luo, Wk Ma, So AMc, Ye Y, Zhang S (2010) Semidefinite relaxation of quadratic optimization problems. IEEE Signal Process Mag 27(3):20–34. https://doi.org/10.1109/MSP.2010.936019
32. Mao S, Chu X, Wu Q, Liu L, Feng J (2021) Intelligent reflecting surface enhanced D2D cooperative computing. IEEE Wirel Commun Lett 10(7):1419–1423. https://doi.org/10.1109/LWC.2021.3069095
33. Guo M, Guan Q, Chen W, Ji F, Peng Z (2022) Delay-optimal scheduling of VMs in a queueing cloud computing system with heterogeneous workloads. IEEE Trans Serv Comput 15(1):110–123. https://doi.org/10.1109/TSC.2019.2920954
34. Wu Q, Zhang S, Zheng B, You C, Zhang R (2021) Intelligent reflecting surface-aided wireless communications: A tutorial. IEEE Trans Commun 69(5):3313–3351. https://doi.org/10.1109/TCOMM.2021.3051897
35. Guo M, Li Q, Peng Z, Liu X, Cui D (2022) Energy harvesting computation offloading game towards minimizing delay for mobile edge computing. Comput Netw 204. https://doi.org/10.1016/j.comnet.2021.108678
36. Yue S, Ren J, Qiao N, Zhang Y, Jiang H, Zhang Y, Yang Y (2022) TODG: Distributed task offloading with delay guarantees for edge computing. IEEE Trans Parallel Distrib Syst 33(7):1650–1665. https://doi.org/10.1109/TPDS.2021.3123535
37. Mahenge MPJ, Li C, Sanga CA (2022) Energy-efficient task offloading strategy in mobile edge computing for resource-intensive mobile applications. Digit Commun Netw 8(6):1048–1058. https://doi.org/10.1016/j.dcan.2022.04.001



Mian Guo (S'11-M'13) received her Ph.D. in communication and information systems from South China University of Technology, China in 2012. She was a visiting professor with the University of Ottawa, Ottawa, Canada in 2016, and a visiting professor with Beihang University, China, in 2017. She is currently an associate professor at Guangdong Polytechnic Normal University, China. Her research interests include resource allocation, QoS provisioning in computer and communication networks, edge computing, and deep reinforcement learning.

Chengyuan Xu received his B.S. degree in Computer Science and Technology from Anyang Institute of Technology, China in 2017. He is currently pursuing a M.S. degree from Guangdong Technical Normal University-Polytechnic Normal University, China. His research interests include edge computing, deep learning and mobile edge computing.

Mithun Mukherjee received the Ph.D. degree in electrical engineering from the Indian Institute of Technology Patna, Patna, India, in 2015. Currently, he is a Professor with the School of Artificial Intelligence, Nanjing University of In- formation Science and Technology, Nanjing, China. Dr. Mukherjee was a recipient of the 2016 EAI WICON, the 2017 IEEE SigTelCom, the 2018 IEEE Systems Journal, and the 2018 IEEE ANTS Best Paper Award. He has been a guest editor for IEEE Internet of Things Journal and IEEE Transactions on Industrial Informatics. His research interests include wireless networks, mobile edge computing, and intelligent edge computing.