Energy Reports 9 (2023) 379–388

Contents lists available at ScienceDirect

Energy Reports

journal homepage: www.elsevier.com/locate/egyr

Minimizing energy consumption of IoT devices for O-RAN based IoT systems☆

Liping Wang [a], Jianhong Zhou [a,c,*], Maode Ma [b], Xianhua Niu [a,c]

[a] School of Computer and Software Engineering, Xihua University, China
[b] College of Engineering, Qatar University, Qatar
[c] National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology, China

ARTICLE INFO

Keywords:
IoT system
Energy consumption
O-RAN
Computation offloading
Non-convex

ABSTRACT

The implementation of Open Radio Access Network (O-RAN) architecture in Internet of Things (IoT) systems has garnered significant attention as a means to fulfill the stringent requirements of ultra-low latency and ultra-low energy consumption in future IoT systems. Although the traditional edge network architecture has been extensively employed, it continues to pose challenges in terms of synchronizing the integration of global and local information for the design of an optimal offloading strategy in edge servers, thus hindering the reduction of latency and energy consumption in IoT devices. In an effort to decrease the latency and energy consumption of IoT devices (IoTDs), we propose a computation offloading problem and employs the Successive Convex Approximation (SCA) algorithm to convert the non-convex problem into a convex problem. The proposed strategy aims to minimize the energy consumption of IoTDs while ensuring Quality of Service (QoS) requirements. The results of the experiment demonstrate that the average energy consumption of terminal devices can be reduced by 20%. Additionally, this strategy is found to be more effective in reducing IoTDs' latency and energy consumption as compared to the traditional edge network architecture.

1. Introduction

Next-generation Internet of Things (IoT) communication networks require real-time connections, ultra-low latency, and massive capacity, and their performance requirements mainly include enhanced mobile broadband (eMBB), ultra-reliable and low-latency communications (uRLLC) and massive machine-type communications (mMTC) (Jones et al., 2020). However, existing wireless architectures lack sufficient flexibility and intelligence to handle these demands effectively. Therefore, it is imminent to transform the architecture to support service heterogeneity, coordination of multi-connection technologies, and on-demand service deployment (Singh et al., 2020). Open Radio Access Network (O-RAN) is an emerging technology that enables such transitions using concepts of virtualization, flexibility, and intelligence. The Operators amalgamated the Cloud Radio Access Network (C-RAN) alliance to form the Open Radio Access Network (O-RAN) alliance, which has the objective of virtualization and intelligence through the incorporation of infrastructure and the integration of embedded Intelligence, as well as providing terminal devices with more expeditious services and improved functionality (O-ran, 2018).

Due to the limited energy, computing, and storage resources of IoT devices in the IoT system, it is difficult for IoT devices (IoTDs) to handle latency-sensitive and computing-intensive applications (eg, smart transportation and smart cities). Therefore, offloading the computing tasks from IoTDs to the edge of the wireless access network, can meet the needs for fast interactive response and provide flexible computing services. It is considered to be an effective solution to make up for the insufficiency of terminal devices' capabilities (Guo et al., 2018). Researchers often design multiple computation offloading strategies based on legacy RAN architectures (eg, C-RAN, Virtual RAN (V-RAN)) to ensure terminal tasks' Quality of Service (QoS) and reduce terminal devices' energy consumption (Muqing and Min, 2019), (Moreira et al., 2021a), (Liang et al., 2021a). However, along with the increasing complexity of the next-generation wireless network, and the traditional RANs' lack of sufficient flexibility and intelligence, the legacy computation offloading strategies are not enough to meet the stringent service requirements in the future IoT systems (Niknam et al., 2020).

In comparison to C-RAN and V-RAN, O-RAN emphasizes openness

☆ 2022 3rd International Conference on Power, Energy and Electrical Engineering (PEEE 2022) 18–20 November 2022.

* Corresponding author at: School of Computer and Software Engineering, Xihua University, China.
E-mail address: zhoujh@uestc.edu.cn (J. Zhou).

https://doi.org/10.1016/j.egyr.2023.09.116
Received 30 August 2023; Accepted 20 September 2023
Available online 14 October 2023

and intelligence as its main aspects (Gavrilovska et al., 2020). First, it realizes the hierarchical management of multi-layer logic functions in RAN. All the logic functions are deployed as virtual functions, which could communicate openly with each other to provide services. Second, the decomposition of near-real-time and non-real-time RICs can support large-scale connectivity for scenarios with high throughput, large coverage, and low power consumption requirements. Finally, the involvement of Distributed Units (DUs) and Central Units (CUs) can provide low-latency support for applications, with high-speed data transmission and large-scale computing requirements.

Therefore, more and more scholars are interested in applying O-RAN to IoT systems (Iturria-Rivera et al., 2022), (Kazemifard and Shah-Mansouri, 2021), (Wang et al., 2022), which aims to better reduce some limitations (eg, resource storage, computing performance, and energy efficiency). Specifically, when computational offloading needs to be performed in IoT devices, it means that the execution of certain tasks can be shifted from IoTDs to the DUs or cloud. The participation and collaboration of DUs and CUs can make precise decisions and optimize allocation. The traditional edge network architecture presents a difficulty in combining global and local information in the edge servers synchronously when performing computational offloading, which leads to legacy offloading strategies being sub-optimal in terms of reducing latency and energy consumption when considering the spatiotemporal in-homogeneity of task arrivals. The deployment of real-time and non-real-time RAN Intelligent Controllers (RICs) enables the collection of both global and local information, respectively, and facilitates communication between them, thus allowing for more precise decisions to be made, resulting in improved congestion reduction and resource conservation. Leveraging these advantages, we seek to enhance the energy efficiency of IoTDs by offloading computation in the Open RAN-based Internet of Things.

In this paper, we endeavor to integrate the O-RAN architecture with the IoT system to render it more intelligent and flexible, wherein edge servers and cloud are established by O-RAN's DUs and O-Cloud, respectively. Non-real-time and near-real-time RICs are deployed in Service Management and Orchestration (SMO) and CU for the joint collection of global and local information, respectively. Meanwhile, in the deployed ORAN based IoT system, a joint optimization computation offloading strategy is proposed to minimize the energy consumption of the IoTDs while meeting the delay requirements of the terminal tasks. The primary contributions of this paper can be divided into the following three points.

1) We consider that local IoTDs, DUs at the edge, and remote O-Cloud can all process computing tasks of IoTDs, and non-real-time and near-real-time RICs deployed in the SMO and CU, respectively, can collect global and local information and communicate with each other through standardized specific interfaces.
2) Based on the deployed ORAN-IoT system, a computation offloading model considering multi-objective costs (i.e. latency and energy consumption) is designed, which are important to trade-off whether the local task needs to be offloaded, done by DUs or O-cloud.
3) In conjunction with the designed computation offloading model, a joint non-convex optimization problem is proposed to minimize the energy consumption of IoTDs by jointly optimizing local processing speed, offloading points, local offloading ratio, and transmission power.

The remainder of the paper is organized as follows: Section 2 introduces preliminary knowledge and related work. Section 3 outlines the main system model. Section 4 defines and analyzes the designed optimization problem. Section 5 presents the simulation results. Section 6 provides a conclusion for the paper.

2. Preliminary knowledge and related work

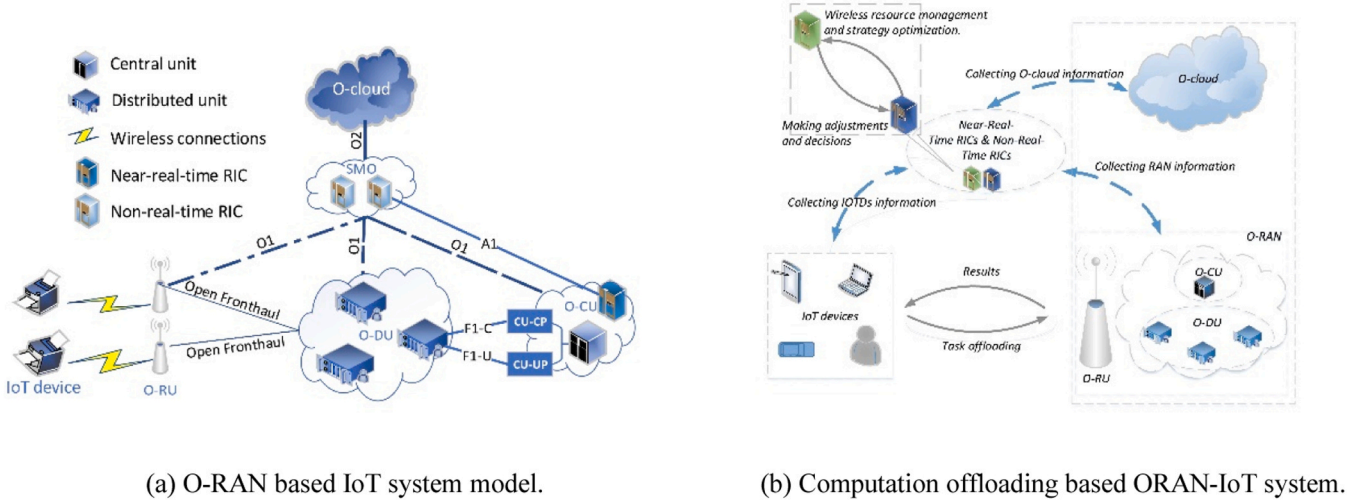
2.1. Preliminary

With the rapid development of the software-defined networking (SDN), the network function virtualization (NFV), the decomposition of dynamic functions, the large-capacity data centers, and cloud computing (Pradhan and Priyanka, 2021), the traditional network architectures have been unable to support various functions and service requirements due to single function and lack of sufficient flexibility. Therefore, the O-RAN comes into being based on the original C-RAN and V-RAN (Cama-Pinto et al., 2021). The C-RAN has been regarded as one of the potential technologies to meet the underlying wireless access requirements in 5 G, which mainly uses the baseband units (BBU) pool for sharing network resources to perform flexible scheduling for efficiency improvements (Mondal and Ruffini, 2022). Owing to the C-RAN supports to software definition and function virtualization, the V-RAN has been gradually developed (Frauendorf and de Souza, É, 2022). The V-RAN increases the scalability and flexibility of the wireless systems on the basis of the C-RAN, and overcomes some drawbacks related to wireless interference and functions. Many great achievements have been made on the C-RAN systems and the V-RAN systems. However, facing the increasing demands and complexities of the wireless networks, the research works on the C-RAN and V-RAN network architectures have gradually shown some limitations.

The features of openness and intelligence have become the theme of the next-generation wireless access networks. And they are also the necessary conditions for the deployment and operation of the next-generation wireless networks (Lagén et al., 2021). Therefore, the O-RANs emerge due to their openness and the intelligence features. Firstly, the O-RAN virtualizes the base stations (BSs) function into network functions, which are further divided into multiple network nodes. The multiple network nodes are the CUs, the DUs, and the RUs, which help to increase the efficiency by executing different network processes. The higher layer of the CUs handles the operations with larger time granularities to implement functions, while the lower layer of the DUs handles time-critical operations, and the RUs manage radio frequency (RF) components and the physical (PHY) layer components (Ranjbar et al., 2022). Secondly, the O-RAN technology separates software, hardware and vendors, who define the open interfaces of the CUs, the DUs, and the RUs by the functions of hardware and software, to improve reliability and availability by modularization and software-based capacity management. Then, the O-RANs can be designed concisely and quickly by extending software, which can reduce the construction cost of the RANs and improve the flexibility of the RANs. Next, the O-RANs can also embed intelligence and extended SDNs to optimize the performance and reduce operational complexity. Finally, the O-RANs divide the CUs into a control plane and a user plane, which can achieve more efficient control and management. The O-RANs introduce RICs including the non-real-time and near-real-time RICs, which allow operators to customize the implementation and deployment of the control plane functions to make better use of resources according to the requirements of operators (Garcia-Saavedra and Costa-Perez, 2021). Until 2021, Japan's Rakuten has deployed a distributed data center for the O-RANs with the units and center functions (Lin, 2021). In addition, some US operators such as Sprint, T-Mobile, Etisalat etc, have also tested and worked on the O-RAN architecture. In the future, more and more suppliers, system integrators and operators will focus on the development of the O-RANs.

2.2. Related work of computation offloading

Recently, computation offloading in different network architectures has received more and more attention, and lots of research work on computation offloading with different network architectures have been carried out (Guglielmi et al., 2018), (Jian et al., 2019), (Moreira et al.,



(a) O-RAN based IoT system model. (b) Computation offloading based ORAN-IoT system.

Fig. 1. (a) O-RAN based IoT system model. (b) Computation offloading based ORAN-IoT system.

2021b), etc. In these studies, some researches focused on designing computing offloading strategy in C-RAN, V-RAN, and MEC, etc, in which effective computation offloading problems are solved together with offloading ratio, computation resource allocation, or radio resource allocation. The authors of (Guglielmi et al., 2018) propose a game-theoretic framework for distributed decision-making when mobile users share the same network resources and no prior information in the wireless link to manage the three-layered computation offloading including mobile devices, edge, and cloud. The authors of (Jian et al., 2019) proposed a joint optimization scheme for offloading and resource allocation in C-RAN based on spectrum efficiency to maximize resource profits. The authors of (Moreira et al., 2021b) proposed a hierarchical software-defined task allocation framework based on V-RAN to effectively determine task offloading strategies while minimizing the transmission time of parallel task execution scheduling. The authors of (Liang et al., 2021b) studied multi-user partial offloading and computing resource management problems in the MEC, who minimized the energy consumption by jointly optimizing transmission delay, local and edge computing capacity allocation, bandwidth allocation and data partitioning. All of the above works are designed to devise different strategies or schemes in the traditional closed or rigid Mobile Edge Computing (MEC) network architecture, however, due to the architectural limitations, this may lead to energy consumption being a bottleneck.

Therefore, the introduction of O-RAN into IoT systems has become a feasible solution for minimizing energy consumption (Pamuklu et al., 2021). In (Pamuklu et al., 2021), the authors have designed a strategy to jointly optimize resource allocation and DUs selection in the O-RAN to reduce network energy consumption and guarantee lower user latency. The works focuses on resource allocation between DUs and CUs, while there is still a lack of relevant research on the cooperation between distributed servers and the cloud in the O-RAN to minimize the energy consumption of IoTDs by making full use of O-RAN characteristics.

3. System model

The designed O-RAN based IoT (ORAN-IoT) system, as depicted in Fig. 1(a), leverages the inherent characteristics of the O-RAN architecture. The IoTDs can transmit task requests to the convergence node RUs, which then forward the requests to the edge servers maintained by the DUs. The majority of physical layer operations are performed by RUs, while high-level protocols are managed by the CUs. The CUs serve as the central point of control for the communication and operation between RUs and DUs, which are connected through high-capacity front-haul links. The CUs have separate data and control planes that are interconnected with the DUs through the F1-U and F1-C interfaces, respectively, in order to optimize hierarchical control and management. The SMO and CU deploy non-real-time and near-real-time RICs, respectively, which collect global and local information through standardized specific interfaces O1 and O2 and communicate through standardized specific interfaces A1 to manage resource allocation and make decisions. The collected global information mainly comprises information regarding the task count, service quality, and DU server availability, which is transmitted to the RICs in the CU for joint analysis and decision-making.

The process of tasks computation offloading is shown in Fig. 1(b). It is assumed that each IoTD has computationally intensive tasks to be completed such as autonomous driving, and collaborative computing. Different tasks have different request quantities, service quality, data size, and computing power requirements. For example, an autonomous driving task has high requirements for service quality and a large number of requests. And a cooperative computing task requires high computing resources because of large amounts of tasks arriving. Although computation offloading can reduce energy consumption and meet the delay requirements of IoTDs, considering the heterogeneity of tasks and the limited resources of the DUs, the design of an optimal computation offloading strategy is a challenging issue. In order to minimize the energy consumption of the IoTDs in the ORAN-IOT system to meet the delay requirement of each task, first of all, the IoTDs request the computation tasks to be offloaded to the O-RAN. Secondly, the non-real-time RICs collect the number of IoTDs, the offloading strategy information, the management information of the CUs, the distance and abilities of the O-cloud, and the available computing resources of the DUs from the RAN and the application servers, which make use of the collecting information to analysis data, manage the non-real-time intelligent wireless resource and optimize the strategy that be deployed on the non-real-time RICs. Then, the non-real-time RICs download the results of analysis, management, and optimization. The near-real-time RICs mainly collect and analyze the real-time variety of the tasks arriving, the requested quantity, the tasks' service quality requirements, and the RAN's resources. Later, the near real-time RICs combine the global information and the optimization strategy provided by the non-real-time RICs to real-time monitor the IoTDs' dynamic changes and make adjustments and decisions by the proposed energy-conserved computation offloading strategy (ECO), which consists of the offloading ratio, the local processing speed and the transmission power of IoTDs, and the decision on the tasks offloaded to the DUs or to the O-cloud to reduce the energy consumption. Finally, only the execution results will be sent back to the IoTDs over the downlink

channel for reducing congestion.

3.1. Channel model

In this ORAN-IoT system design, it is postulated that there exist $m(m \in \{1, 2, ..., M\})$ IoT terminal devices. These devices are capable of sending task requests to RUs through wireless transmission. In light of this, the issue of inter-RU interference must be taken into account, as we are contemplating the multiplexing of the full frequency spectrum, which is overlaid by different RUs. Conversely, the issue of intra-RU interference can be disregarded, as the spectrum is orthogonally allocated to IoT devices when accessing the same RU. It is further assumed that all downlink connections between RUs and IoT devices possess complete Channel State Information (CSI). As a result, the Signal to Interference to Noise Ratio (SINR) transmitted by RU r to IoT device m can be evaluated.

$$SINR_{m,r} = \frac{p_m g_{m,r}}{\sigma^2 + \sum_{j=1, j\neq r}^{R} p_m g_{m,j}}, \# \tag{1}$$

where the channel gain of the transmission from RU r to IoT device m is represented by $g_{m,r}$, and the transmitting power of IoT device m is denoted by p_m. The presence of Additive Gaussian White Noise (AGWN) is characterized by the power represented by σ^2, which follows a Gaussian distribution. Consequently, the propagation rate of RU r to IoT device m can be described by the Shannon formula as

$$V_{m,r} = B\log_2\left(1 + \frac{p_m g_{m,r}}{\sigma^2 + \sum_{j=1, j\neq r}^{R} p_m g_{m,j}}\right), \# \tag{2}$$

where the total bandwidth is denoted by B.

3.2. Delay model

In the present system, the predominant sources of latency are comprised of queuing, processing, and transmission delays. The queuing latency at the edge DUs is represented by Q_m^D, while the queuing latency at the remote O-cloud is not considered as it boasts ample computing resources and therefore does not necessitate queuing. The transmission latency, which primarily involves the delay incurred during the transmission of terminal tasks to either the O-cloud or DUs, is denoted by $t_{m,off}^C$ and $t_{m,off}^D$, respectively. The processing latency, resulting from the processing of tasks in the DUs, O-cloud, and local IoT device, is expressed as $t_{m,exe}^D$, $t_{m,exe}^O$, and $t_{m,exe}^L$, respectively. Subsequently, a modeling approach for these three types of delays will be proposed.

3.2.1. The queuing latency

As previously stated, the consideration of queuing delays is limited to the distribution units DUs. To model the queuing latency for each DU, we adopt the M/M queueing model with one DU server. The arrival of terminal tasks is assumed to follow a Poisson distribution, represented by the first 'M′, while the rate of service is assumed to be exponentially negatively distributed, represented by the second 'M′. Thus, the queuing delay at the DU can be expressed as follows

$$Q_m^D = \frac{1}{Y_d - X_d}.\# \tag{3}$$

The average rate of task arrival at the DU is represented by Y_d, while the average rate of task servicing at the DU is denoted by X_d.

3.2.2. The processing latency

The computing capabilities assigned to tasks from the O-cloud, DUs, and local IoT devices are represented by f_m^C, f_m^D, and f_m^L, respectively. Consequently, the processing latency for each of these entities can be expressed as follows:

$$t_{m,exe}^C = \frac{(1-\lambda_m)D_m}{f_m^C}, \# \tag{4}$$

$$t_{m,exe}^D = \frac{(1-\lambda_m)D_m}{f_m^D}, \# \tag{5}$$

$$t_{m,exe}^L = \frac{(1-\lambda_m)D_m}{f_m^L}, \# \tag{6}$$

3.2.3. The transmitting latency

It is assumed that the task data generated by the m_{th} local IoT device, denoted by D_m, is of a certain size, and the proportion of processing that takes place in the local IoT device is represented by λ_m, with the constraint $0 < \lambda_m < 1$. As a result, the task data that is offloaded to the O-cloud or DUs is calculated as $(1-\lambda_m)D_m$. Furthermore, it is assumed that the transmission capacity of the high-capacity front-haul link between the RU r and DU d is represented by $V_{r,d}$. Based on these assumptions, the varying transmission delays can be mathematically expressed as

$$t_{m,off}^C = \frac{(1-\lambda_m)D_m}{V_{m,r}} + d_m, \# \tag{7}$$

$$t_{m,off}^D = \frac{(1-\lambda_m)D_m}{V_{m,r}} + \frac{(1-\lambda_m)D_m}{V_{r,d}}, \# \tag{8}$$

where the transmitting latency of wired transmission from DUs to the O-cloud is denoted by d_m.

3.3. Energy model

In this system, the primary energy consumption, represented by e_m, encompasses the energy consumption incurred during both processing and transmitting. The processing energy consumption is restricted to the local IoT devices and can be represented by e_m^L. The transmission energy consumption, on the other hand, encompasses the tasks transmitted to the DUs and O-cloud, which are denoted by e_m^D and e_m^C, respectively. Thus, the various energy consumption values, e_m^L, e_m^D, and e_m^C, can be calculated as follows

$$e_m^D = p_m t_{m,off}^D + r_m, \# \tag{9}$$

$$e_m^C = p_m t_{m,off}^C + r_m, \# \tag{10}$$

$$e_m^L = \varepsilon \lambda_m D_m {f_m^L}^2, \# \tag{11}$$

where the energy consumption consumed by the channel after the task transmitted is expressed as r_m, and the hardware factor is denoted by ε.

3.4. Offloading model

In this section, two cost indicators are proposed to balance the trade-off between delay and energy consumption when offloading the tasks of IoT devices to the O-cloud and DUs. The cost indicators correspond to the energy consumption and delay costs associated with offloading to the O-cloud and DUs, respectively. For IoT device m, the offloading selection is formulated as a binary variable, denoted by a_m, such that if the cost of offloading to the DUs is lower, a_m is equal to 0; otherwise, if the cost of offloading to the O-cloud is lower, a_m is equal to 1. The total delay experienced by IoT device m when offloaded to the DUs and O-cloud can be obtained as t_m^D and t_m^C, respectively.

$$t_m^D = t_{m,off}^D + t_{m,exe}^D + Q_m^D, \# \tag{12}$$

$$t_m^C = t_{m,off}^C + t_{m,exe}^C, \# \tag{13}$$

Consequently, the cost associated with the delay and energy consumption of offloading to the O-cloud and DU, represented by L_m^C and L_m^D, respectively, can be determined through the following expressions.

$$L_m^D = \gamma_m^e e_m^D + \gamma_m^t t_m^D, \# \tag{14}$$

$$L_m^C = \gamma_m^e e_m^C + \gamma_m^t t_m^C, \# \tag{15}$$

where, γ_m^e , $\gamma_m^t \in [0,1]$, which means that the weight of energy consumption and delay can be adjusted according to the task and device requirements.

Based on the formulation described above, when the cost corresponding to the delay and energy consumption of offloading to the O-cloud and DU are equal, i.e., $L_m^C = L_m^D$, a threshold d_{thr} associated with d_m can be computed using a simplified expression as follows

$$d_{thr} = \frac{D_m\left(f_m^C - f_m^D\right)}{\left(1 + \frac{p_m \gamma_m^e}{\gamma_m^t}\right) f_m^C f_m^D}. \# \tag{16}$$

When $a_m = 1$, which indicates $d_m < d^{thr}$, and $a_m = 0$, which indicates $d_m > d^{thr}$.

4. Efficient energy consumption for computation offloading

In this section, the various models described above are integrated to formulate a joint non-convex optimization problem aimed at minimizing the energy consumption of IoT devices. The optimization problem considers the joint optimization of local processing speed, offloading points, local offloading ratio, and transmission power. To effectively address the non-convex nature of the problem, the original problem is transformed into a convex problem through a mathematical transformation, thereby enabling the realization of an energy-efficient computation offloading strategy.

4.1. Problem formulation

According to the modeled delay and energy consumption model, for the m_{th} device, the total delay and energy consumption can be calculated as follows, respectively.

$$e_m = e_m^L + a_m e_m^C + (1 - a_m) e_m^D, \# \tag{17}$$

$$t_m = \max\left\{t_{m,exe}^L, a_m t_m^C + (1 - a_m) t_m^D\right\}. \# \tag{18}$$

Our objective is to minimize the energy consumption of IoT devices by jointly considering the offloading selection $A = (a_m) \forall m$, the local IoTDs' computation speed $F^L = \left(f_m^L\right) \forall m$, the offloading ratio $\lambda = (\lambda_m) \forall m$, and the offloading transmission power $P = (p_m) \forall m$. The energy consumption of all IoTDs, $\sum_{m=1}^{M} \omega e_m$, is balanced through a weighted summation of the energy consumption based on the task and device requirements. Hence, the computation offloading problem that takes into account A, λ, f^L, and P, can be formulated as follows:

$$\begin{aligned}
&\mathrm{P}_1: \min_{f^L,\, A,\, P,\, \lambda} \quad \sum_{m=1}^{M} \omega e_m \\
&s.t: \\
&C_1: 0 \le p_m \le P_{MAX}, \forall m \\
&C_2: 0 \le f_m^L \le F^L, \forall m \\
&C_3: 0 \le t_m \le T_{MAX}, \forall m \\
&C_4: 0 \le \lambda_m \le 1, \forall m \\
&C_5: \quad a_m \in [0,1], \forall m
\end{aligned}$$

Subject to the constraints C_1, C_2, C_3, C_4, and C_5, which specify the offloading transmitting power constraint of IoTD m's task, the maximum local computation speed constraint, the maximum latency constraint of IoTD m's task, the offloading ratio constraint of IoTD m's task, and the offloading selection constraints, respectively. The problem, represented as P_1, is non-convex in nature and hence, difficult to solve directly. However, the complexity of the original problem is reduced by first decoupling it into two sub-problems and then using the Successive Convex Approximation (SCA) algorithm to approximate it to a convex problem, ensuring that the original problem P_1 can be solved iteratively through iteration over the two sub-problems.

4.2. Problem analysis

In order to minimize the complexity of Problem P_1, a reduction in the dimensionality of the problem is necessary. In regard to the local computation speed, f^L, it is evident that it has a proportional relationship with the local processing energy consumption, as demonstrated in Eq. (11). Thus, f^L can be transformed into equations related to λ_m and T_{MAX} by maximizing the local task latency constraint, as represented in the following transformation.

$$\frac{\lambda_m D_m}{T_{MAX}} \le f_m^L, \# \tag{19}$$

and

$$0 \le \lambda_m \le \min\left\{1, \frac{F^L T_{MAX}}{D_m}\right\}. \# \tag{20}$$

In this paper, the computation offloading model employs two metrics, energy consumption and delay, to determine the offloading destination of the tasks executed by IoTD m. To simplify the complexity of the original problem P_1, we decouple the solution of the variable A into a separate problem, designated as P_2. As indicated by Eq. (16), the variables D_m and p_m are related to the value of A. Therefore, given fixed values of D_m and p_m, the value of A can be determined by considering the known variables f_m^C, f_m^D, and f_m^L. The formulation of P_2 can be represented as follows:

$$\begin{aligned}
&\mathrm{P}_2: \underset{f_m^L, f_m^D, f_m^C, D^*, p^*}{\text{search}} \quad \mathrm{A} \\
&\quad \text{s.t: } \mathrm{C}_3, \mathrm{C}_5 \\
&\mathrm{C}_6: \frac{\lambda_m D_m}{T_{MAX}} \le f_m^L \le F^L, \forall m \\
&\mathrm{C}_7: 0 \le f_m^D \le F^D, \forall m \\
&\mathrm{C}_8: 0 \le f_m^C \le F^C, \forall m \\
&\mathrm{C}_9: D^* \le D_m, \forall m \\
&\mathrm{C}_{10}: p^* \le P_{MAX}, \forall m
\end{aligned}$$

where C_6 is obtained from C_2 and (19), C_7 is the maximum DUs processing speed constraints, C_8 is the maximum O-cloud processing speed constraints, C_9 and C_{10} are the maximum number of task of IoTD m and the maximum transmit power constraints, respectively. Then, search for the value of A according to P_2 which can be fixed as a_m^*. Thus, Problem P_3 can be transformed by combining problem P_1 and P_2, which is expressed as:

$$\begin{aligned}
&P_3: \min_{P, \lambda} \quad \sum_{m=1}^{M} \omega e_m \\
&\text{s.t: } C_1. \\
&C_{11}: 0 \le \max\{\max\{t_{m,exe}^L, a_m^* t_m^C + (1 - a_m^*) t_m^D\}\} \le T_{MAX}, \forall m \\
&C_{12}: 0 \le \lambda_m \le \min\{1, \tfrac{F^L T_{MAX}}{D_m}\}, \forall m
\end{aligned}$$

The constraints C_{11} and C_{12} are obtained from the transformation of C_3 and C_4, respectively, representing the delay constraint and the offloading ratio constraint after the reduction of variables A and f^L. When the value of a_m is fixed at a_m^*, indicating that the offloading selection is

established, the processing delay at the DUs and O-cloud, as well as the queuing delay at the DUs, can be fixed to t_m^*. As a result, C_{13} can be re-expressed as follows:

$$h_m(p_m,\lambda_m): \frac{(1-\lambda_m)D_m}{B\log_2\left(1+\frac{p_m g_{m,r}}{\sigma^2+\sum_{j=1,j\neq r}^{R} p_m g_{m,j}}\right)} - T_{MAX} + t_m^* \le 0.\# \tag{21}$$

The Eq. (21) presents a challenge in terms of its non-convex nature, making it difficult to solve. To overcome this difficulty, the non-convex portion can be separated from Eq. (21) and addressed in subsequent steps. Thus, Eq. (21) can be split into Eqs. (22) and (23), based on their convexity and non-convexity, respectively.

$$Convex: h_m^{'}(\lambda_m) = \frac{B_m\left(t_m^* - T_{MAX}\right)}{(1-\lambda_m)D_m}.\# \tag{22}$$

$$non-convex: h_m^{'}(p_m) = -\log_2\left(1+\frac{p_m g_{m,r}}{\sigma^2+\sum_{j=1,j\neq r}^{R} p_m g_{m,j}}\right).\# \tag{23}$$

For the non-convex component of formula (23), we resort to the successive convex approximation (SCA) technique to solve a sequence of convex substitution functions of the original problem and eventually converge towards the stationary solution of the original problem (Marks and Wright, 1978). The central idea behind SCA is to approximate the non-convex problem and by iteratively finding a sub-optimal solution that is closely approximated to the optimal solution. Thus, we assume that $p_m = 2^{q_m}$, and formula (23) can be re-expressed as

$$h_m^{'}(q_m) = -\log_2\left(1+\frac{q_m g_{m,r}}{\sigma^2+\sum_{j=1,j\neq r}^{R} q_m g_{m,j}}\right).\# \tag{24}$$

In the successive convex approximation (SCA) process, each convex approximation of $h_m^{'}(q_m^k)$ must abide by the following three properties, as demonstrated below:

$$h_m^{'}\left(q_m^{(k)}\right) \le \tilde{}_{h_m}{}'\left(q_m^{(k)};q_m^{(k-1)}\right), \forall m,k\# \tag{25}$$

$$h_m^{'}\left(q_m^{(k-1)}\right) = \tilde{}_{h_m}{}'\left(q_m^{(k-1)};q_m^{(k-1)}\right), \forall m\# \tag{26}$$

$$\nabla h_m^{'}\left(q_m^{(k)}\right) = \nabla\tilde{}_{h_m}{}'\left(q_m^{(k-1)};q_m^{(k-1)}\right), \forall m\# \tag{27}$$

where $\tilde{}_{h_m}{}'(q_m^{(k)};q_m^{(k-1)})$ indicates $h_m^{'}(q_m)$ at the k_{th} sequence of convexification.

$$\tilde{}_{h_m}{}'(q_m^{(k)};q_m^{(k-1)}) \triangleq -\varphi_m^{(k-1)}\log_2(g_m) + q_m^{(k)} - \log_2(\sigma^2+\sum_{j=1,j\neq r}^{R} q_m g_{m,j})) - \beta_m^{(k-1)}, \forall m\# \tag{28}$$

The formulation (25) means that $h_m^{'}(q_m^{(k)})$ does not need to be its own upper bound at any feasible point, so we have

$$h_m^{'}(q_m^{(k)}) \le -\varphi_m^{(k-1)}\log_2(g_m) + q_m^{(k)} - \log_2(\sigma^2+\sum_{j=1,j\neq r}^{R} q_m g_{m,j})) - \beta_m^{(k-1)}, \forall m\# \tag{29}$$

$$\varphi_m^{(k-1)} \triangleq \frac{S_m^{(k-1)}}{1+S_m^{(k-1)}}\# \tag{30}$$

$$\beta_m^{(k-1)} \triangleq \log_2\left(1+S_m^{(k-1)}\right) - \frac{S_m^{(k-1)}}{1+S_m^{(k-1)}}\log_2\left(S_m^{(k-1)}\right),\# \tag{31}$$

$$S_m^{(k-1)} \triangleq \frac{2^{q_m^{(k-1)}} g_{m,r}}{\sigma^2+\sum_{j=1,j\neq r}^{R} 2^{q_m^{(t-1)}} g_{m,j}}, \forall m\# \tag{32}$$

Therefore,

$$h_m\left(q_m^{(k)},\lambda_m^{(k)}\right) \le -\varphi_m^{(k-1)}\left(\Gamma_m + q_m^{(k)}\right) - \beta_m^{(k-1)} + \frac{B\left(t_m^* - T_{MAX}\right)}{\left(1-\lambda_m^{(k)}\right)D_m} \le 0, \forall m\# \tag{33}$$

where

$$\Gamma_m \triangleq \log_2(g_m) - \log_2(\sigma^2+\sum_{j=1,j\neq r}^{R} 2^{q_m^{(t-1)}} g_{m,j})).\forall m\# \tag{34}$$

After the k_{th} sequence of convexification, a suitable approximation for $h_m^{'}(q_m)$ could be found. We can convert P_3 to P_4, which can be expressed as follows:

$$P_4: \min_{q,\lambda} \quad \sum_{m=1}^{M} \omega e_m$$

s.t:

$$C_{13}: -\varphi_m(\Gamma_m+q_m) - \beta_m + \frac{B_m(t_m^* - T_{MAX})}{(1-\lambda_m)D_m} \le 0, \forall m$$

$$C_{14}: \lambda_m - 1 \le 0, \forall m$$

C_{13} is transformed by (33), and C_{14} is similar to C_4. Subsequently, the convex problem is solved utilizing the Lagrange multiplier method, in order to derive a closed-form expression in the Karush-Kuhn-Tucker (KKT) condition. The KKT condition is presented as follows

$$L(q_m,\lambda_m,\vartheta_m,\mu) = \omega e_m + \sum_{m=1}^{M}\vartheta_m[-\varphi_m(\Gamma_m+q_m) - \beta_m + \frac{B\left(t_m^* - T_{MAX}\right)}{(1-\lambda_m)D_m}] + \sum_{m=1}^{M}\mu(\lambda_m - 1),\# \tag{35}$$

The k_{th} iteration point is $q_m^{(k)}$, by which a surrogate function of $h_m^{'}(q_m)$ can be constructed. As the k_{th} sequence of convexification retains the geometric features of $h_m^{'}(q_m)$ locally in $q_m^{(k)}$, and are strongly convex, which can be expressed as

and

$$\frac{\partial L}{q_m} = (\log 2)2^{q_m} - \vartheta_m\varphi_m = 0,\# \tag{36}$$

$$\frac{\partial L}{\lambda_m} = \frac{-\left(t_m^* - T_{MAX}\right)}{(1-\lambda_m)^2 D_m} + \mu = 0.\# \tag{37}$$

Therefore, based on the above solution, equations can be established for the offloading transmission power, p_m, and the offloading ratio, λ_m. Subsequently, the optimal offloading ratio, λ_m, and transmit power, p_m, for IoTD m can be determined and expressed as follows:

$$\lambda_m = 1 - \sqrt{\frac{t_m^* - T_{MAX}}{\mu D_m}}.\# \tag{38}$$

$$p_m = 2^{\left[\frac{1}{\varphi_m}\left(-\beta_m + \frac{B(t_m^* - T_{MAX})}{(1-\lambda_m)D_m}\right) - \Gamma_m\right]}.\# \tag{39}$$

Algorithm 1. Joint energy computation offloading algorithm to solve P_1.

Input: $B, \gamma_m^e, \gamma_m^t, \lambda_m^{(0)}, p_m^{(0)}, T_{MAX}$

Output: p_m and λ_m

Minimizing local computation speeding f_m^L from (19)

Searching A by computing the value of d_{thr} by using (14), (15), (16)

Initialize $k \leftarrow 1$

Repeat

Update $p_m^{(k)}$ according SCA algorithm

$k \leftarrow k + 1$

until (21) find a suitable value of convex approximation

Update A

Until convergence

4.3. Overall algorithm

The overall design of the proposed optimization scheme is divided into two parts including the search for the solution offloading point a_m and the allocation of the transmit power p_m with the offloading ratio λ_m based on the predetermined offloading point, and finally minimize IoTDs' energy consumption by iterating the two portions. Specifically, the IoTD m accesses the RU r, with the initialization of the offloading ratio $\lambda_m^{(0)}$ and transmit power $p_m^{(0)}$. At the beginning of each iteration, the value of the offloading option a_m is searched with an initial offloading ratio and a transmit power. Then, according to the value of A, the offloading ratio and the transmit power will be updated alternately and iteratively by solving Eq. (21). Finally, the corresponding value of A, transmit power p_m and offloading ratio λ_m will be obtained. The energy consumption will be also obtained at the same time. When the energy consumption is no longer reduced and the constraint C_{12} is satisfied, then the algorithm ends.

Using the convex approximation and the Lagrange multiplier method, the offloading ratio and transmit power allocation problems are jointly solved by P_4. First, the value of A is updated according to the initialized offloading ratio $\lambda_m^{(0)}$ and transmit power $p_m^{(0)}$. To satisfy the establishment of the loop condition, the time frequency is $N + 1$. Then, the formula (21) is solved convexly by nested loop as shown in Algorithm 1, because the loop body is only executed N times, the time frequency is $N(N+1)$ and the time frequency of the nested loop body is N^2. Finally, the transmit power p_m and offloading ratio λ_m are updated according to the convexization results. Therefore, according to the sum of the time frequency is $2N^2 + 2N + 1$, which can be concluded that the time complexity of the whole algorithm is $O(N^2)$ and the computational complexity is relatively low.

5. Simulation results

In this section, Matlab is used to implement a simulation experiment for the computation offloading procedure in the ORAN-IoT system by adopting numerical simulation, to show the effectiveness of the proposed ECO strategy. We randomly deploy 30 IoTDs and 3 DUs. All the IoTDs are wirelessly connected to their adjacent DUs, which are wired connected to each other and connected to the O-cloud. The partial simulation parameters are summarized as follows in Table 1.

Our proposed ECO strategy is evaluated by comparing its average energy consumption with that of four alternative strategies (Guglielmi et al., 2018; Liang et al., 2021b; Sun et al., 2019; Alahmadi et al., 2020). The energy consumption is analyzed in terms of the energy used for processing within IoT devices and the energy consumed during offloading tasks from these devices. The first strategy, without offloading (WOO), involves solely local computation of IoT tasks. The second strategy, without leaving offloading (WLO), balances local computation and offloading through weighing in between minimizing energy consumption and latency. The third strategy, the legacy-offloading (LO), implements legacy offloading by jointly considering the energy consumption, latency, and the amount of the data. The fourth and final strategy, ECO, which is the strategy proposed in this study, considers various factors including local processing speed, offloading points, local offloading ratio, and transmission power in order to optimize the energy conservation during offloading.

As shown in Fig. 2(a), it is clear that the offload-based LO, WLO, and ECO strategies of average energy consumption outperform the local execution-based WOO strategy. The reason is that all the tasks of the WOO strategy are executed on the IoTDs, although there is no offload

Table 1
Partial Parameters.

Parameter	Typical Value
σ^2	$2*10^{-13}$
$C_{r,d}$	100 MHZ
γ_m^e	0.5
γ_m^t	0.5
P_{MAX}	0.1 W
T_{MAX}	1 s
F^L	0.5 GHz

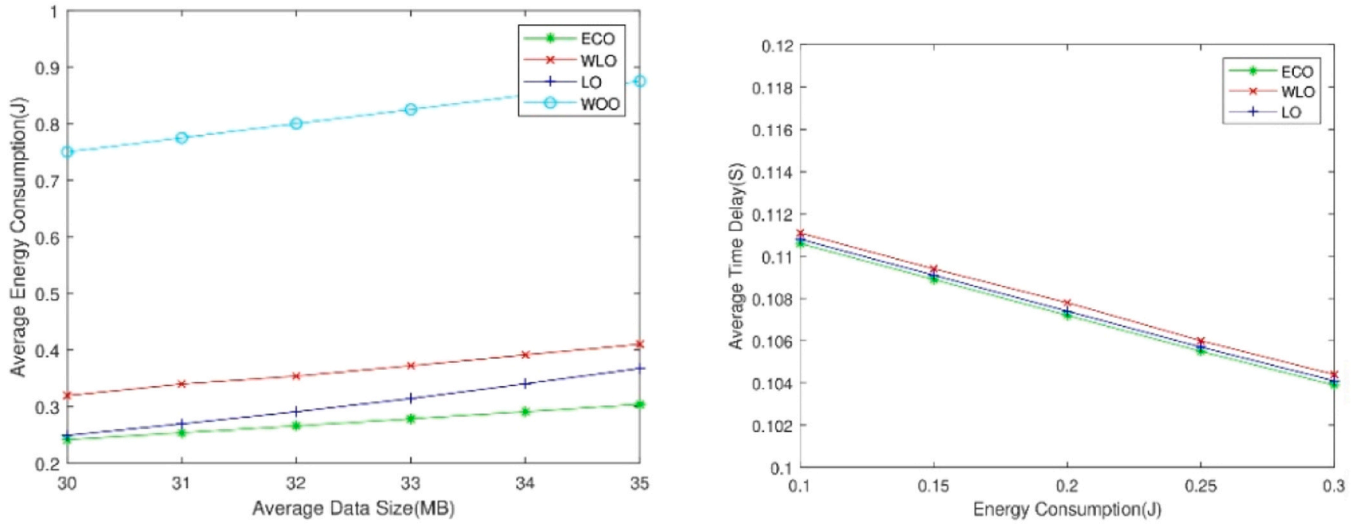


Fig. 2. (a) The average energy consumption comparison with the number of tasks under four strategies; (b) Delay comparison with the same energy consumption under different strategy.

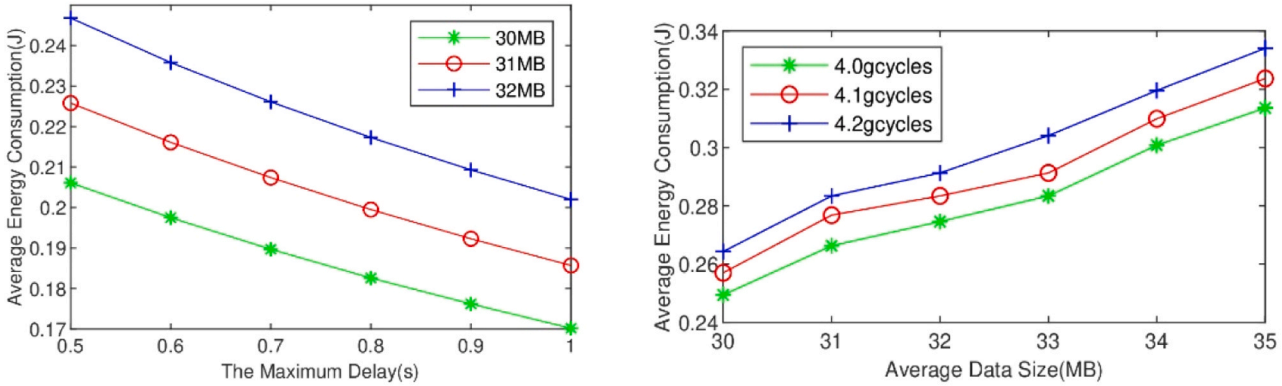


Fig. 3. (a) The average energy consumption comparison with different delay constraints; (b) The average energy consumption comparison with different local processing speed.

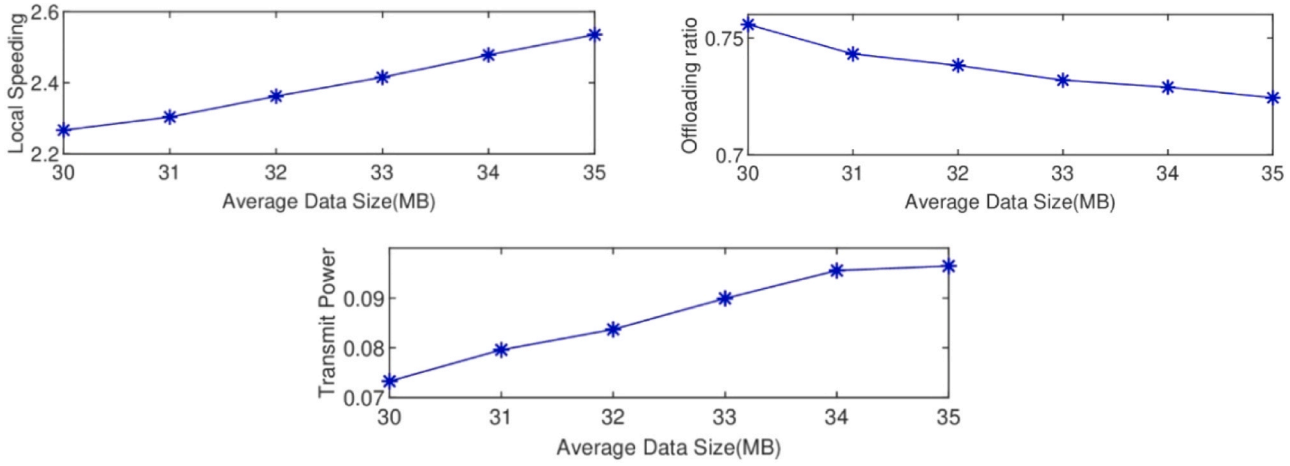


Fig. 4. The various of local processing speed, transmission power, and offloading ratio.

energy consumption, the amount of local processing tasks is too large, resulting in extremely high local processing energy consumption. And for the offloading-based strategy, the local executing and transmitting energy consumption can be better balanced by offloading tasks, thereby reducing the total energy consumption. At the same time, as the number of tasks increases, the average energy consumption of the ECO strategy is the lowest compared with the WLO strategy and the LO strategy. The reason is that the WLO strategy only considers the trade-off between local computation and task offloading, and to satisfy the task's maximum delay constraint, a larger local computation speed and higher transmit power are required, which will lead to the local and transmit energy consumption increase. For the LO strategy, the tasks are

offloaded based on a designed offloading model that jointly considers energy consumption and delay. Although local processing and task offloading can be better balanced, local computing power is not fully utilized due to the inflexible adjustment of local speed, which leads to local energy consumption cannot be better reduced. The proposed ECO strategy that based on the designed offloading model considering energy consumption and delays cost, which can flexibly coordinate local processing speed, offload rate, and transmit power, and can better optimize local processing and transmission under the premise of satisfying delay constraints to reduce the average energy consumption. Therefore, it is obvious that the designed ECO strategy is the most effective of the four strategies in energy saving, which is 26% and 12% lower than that of the WLO strategy and the LO strategy, respectively.

Since latency is one of the most important optimizable indicators in computation offloading problems, we compare the propagation delays of the three strategies executed under the same energy consumption conditions as shown in Fig. 2(b). The reason why the transmission delay of the WOO strategy is not compared is that the WOO strategy is only processed locally, which means that the transmission delay is not involved in the WOO strategy. It can conclude from Fig. 2(a) that if an equal number of tasks arriving, the increase in energy consumption can reduce the latency of the three offloading strategies. And the increase in energy consumption means that the local device can have a higher processing speed or higher transmit power, which means that more tasks can be processed locally to avoid being offloaded or can have higher transmit power to transmit, to reduce the transmitting delay effectively. Therefore, the ECO strategy can flexibly coordinate local processing speed, offloading rate, and transmission power, under the same energy consumption, the transmission delay consumed by the ECO strategy is always the lowest among the three strategies.

Another experiment has the goal to minimize the average energy consumption by meeting the requirement of the maximum delay. Therefore, Fig. 3(a) shows the average energy consumption under the maximum delay constraints T_{MAX} with the increase of the number of tasks arriving. It is obviously that the average energy consumption of the lower delay constraint is higher than that of the higher delay constraint and the average energy consumption of the larger the number of tasks arriving is also higher. The reason is that the smaller the T_{MAX}, the stricter the delay constraint, which indicates that fewer IoTDs are suitable for offloading tasks to the DUs or to the O-cloud. So as to meet the requirements of delay when the number of tasking arriving, more IoTDs are needed to improve the local speeding and to increase the transmission power and the energy consumption of the IoTDs. In addition, when the maximum delay constraints are small, there will be more delay-sensitive IoTDs. For the delay-sensitive IoTDs, the allocation of a large number of wireless resources during the offloading process makes the co-channel interference based on the reuse frequency will be more serious, and will also increase the energy consumption. At the same time, Fig. 3(a) demonstrates that the effectiveness of the ECO strategy to reduce the energy consumption by adjusting the offloading ratio, local speed and transmission power.

The ECO strategy calculates the local processing speed by considering the local number of tasks and the maximum task delay constraints. The local computation speeding is usually determined by the hardware devices. According to the formulation (11), it is clear that the local speed is directly proportional to the energy consumption of the IoTDs. The higher the local speeding, the higher the energy consumption. According to formulation (19), it is clear to see that the greater the local speed, the lower the latency of processing tasks. Therefore, in order to reduce the energy consumption under the condition to meet the task delay requirements, for the non-computationally intensive IoTDs tasks, they are more inclined to process tasks at the local devices, which needs to minimize the local speed to reduce the energy consumption. For the intensive IoTDs tasks, due to the limited local speeding, if the tasks are processed at the local devices, the processing delay will increase. With the aim to meet the delay requirements, the tasks have to be offloaded to the DUs or to the O-cloud to make the energy consumption minimized by adjusting the local speed, the offloading ratio and the transmission power by the ECO strategy. Thence, in Fig. 3(b), we compare the average energy consumption with the increase of the number of tasks arriving at the different local processing speed. It is assumed that the tasks arriving is intensive, which implies that the tasks have to be offloaded. It can be seen that at the same number of tasks arriving, with the local speeding increases, the energy consumption will increase. At the same local speed, with the number of tasks arriving increase, the energy consumption will increase. In addition, the energy consumption gap among the three different local speeds verifies the impact of the offloading decision and performance by adjusting the local processing speed.

We mentioned earlier that the energy consumption is directly related to the local processing speed, the transmission power, and the offloading rate. Therefore, Fig. 4 shows the changes of the local processing speed, the transmission power and the offloading ratio during the offloading process, respectively. As the number of task arriving increases, the local processing speed and the transmission power will gradually increase, the offloading ratio λ will gradually decrease, which means that the $1-\lambda$ will gradually increase and more larger ratio of task is offloaded. It shows that the IoTDs are willing to offload more tasks to reduce the energy consumption. The local speed is relevant to the maximum delay T_{MAX} of the task and the $1-\lambda$, which illustrates that the tasks need to be processed at the local devices, which shows an increasing trend due to the reason that the local processing speed will gradually increase. In order to improve the transmission rate to meet delay requirement, the transmission power will increase with the number of tasks offloaded increase. However, due to the limitation of the maximum power, it will gradually approach 0.1w and will not exceed the value. Owing to the limitation of the delay of the tasks and the local processing speed, the offloading ratio λ will be decreased and the $1-\lambda$ will increase, which indicates that the proportion of being processed locally is gradually reduced, and more and more tasks prefer to be offloaded to the O-cloud or DUs.

6. Conclusion

The integration of the O-RAN architecture within the IoT system has garnered significant attention as a research hotspot, given the potential to address performance limitations such as energy consumption, delay, and resource utilization in traditional IoT systems. In this paper, we present a flexible offloading strategy for deployed IoT systems and formulate a non-convex optimization problem with the objective of minimizing energy consumption while ensuring task delay requirements are met. To tackle the non-convex nature of the problem, the SCA algorithm is employed, allowing for its approximate conversion into a solvable convex problem. Results from simulations demonstrate that the proposed offloading strategy significantly reduces the energy consumption of IoT devices. In the future, the O-RAN architecture will be leveraged to intelligently manage communication and computing resources, contributing to the enhancement of the overall performance of the IoT system and meeting evolving requirements.

Declaration of Competing Interest

The authors have no conflicts of interest to disclose and all authors have approved the manuscript for submission.

Data availability

The data that has been used is confidential.

Acknowledgements

The work of this paper was supported by the National Science Foundation of China (No. 62171387).

References

A.A. Alahmadi T.E. El-Gorashi J.M. Elmirghani Energy efficient and delay aware vehicular edge cloud 2020 22nd International Conference on Transparent Optical Networks ((ICTON)). IEEE 2020 1 4.

Cama-Pinto, D., Damas, M., Holgado-Terriza, J.A., Gómez-Mula, F., Calderin-Curtidor, A. C., Martínez-Lao, J., Cama-Pinto, A., 2021. 5g mobile phone network introduction in Colombia. Electronics 10 (8), 922.

Frauendorf, J.L., de Souza, É, Almeida, 2022. The Evolution of RAN (Radio Access Network), D-RAN, C-RAN, V-RAN, and O-RAN. In The Architectural and Technological Revolution of 5G. Springer International Publishing,, Cham, pp. 139–154.

Garcia-Saavedra, A., Costa-Perez, X., 2021. O-RAN: disrupting the virtualized RAN ecosystem. IEEE Commun. Stand. Mag. 5 (4), 96–103.

Gavrilovska, L., Rakovic, V., Denkovski, D., 2020. From cloud ran to open ran. Wirel. Pers. Commun. 113 (3), 1523–1539.

Guglielmi, A.V., Levorato, M., Badia, L., 2018. A Bayesian game theoretic approach to task offloading in edge and cloud computing (May)In 2018 IEEE International Conference on Communications Workshops (ICC Workshops) 1 6.

Guo, H., Liu, J., Zhang, J., Sun, W., Kato, N., 2018. Mobile-edge computation offloading for ultra-dense iot networks. IEEE Internet Things J. 5 (6), 4977–4988.

Iturria-Rivera, P.E., Zhang, H., Zhou, H., Mollahasani, S., Erol-Kantarci, M., 2022. Multi-agent team learning in virtualized open radio access networks (o-ran). Sensors 22 (14), 5375.

Jian, Z., Muqing, W., Min, Z., 2019. Joint computation offloading and resource allocation in C-RAN with MEC based on spectrum efficiency. IEEE Access 7, 79056–79068.

Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B., 2020. Characterising the digital twin: a systematic literature review. CIRP J. Manuf. Sci. Technol. 29, 36–52.

Kazemifard, N., Shah-Mansouri, V., 2021. Minimum delay function placement and resource allocation for open ran (o-ran) 5g networks. Computer Netw. 188, 107809.

Lagén, S., Giupponi, L., Hansson, A., Gelabert, X., 2021. Modulation compression in next generation RAN: air interface and fronthaul trade-offs. IEEE Commun. Mag. 59 (1), 89–95.

Liang, B., Fan, R., Hu, H., 2021a. Energy-efficient task offloading and resource allocation for multiple access mobile edge computing. arXiv preprint arXiv 2103, 03740.

Liang, B., Fan, R., Hu, H., 2021b. Energy-efficient task offloading and resource allocation for multiple access mobile edge computing. arXiv preprint arXiv 2103, 03740.

Lin, B.S., 2021. Toward an ai-enabled o-ran-based and sdn/nfv-driven 5g& iot network era. Netw Commun. Technol. 6 (1), 6–15.

Marks, B.R., Wright, G.P., 1978. A general inner approximation algorithm for nonconvex mathematical programs. Oper. res. 26 (4), 681–683.

Mondal, S., Ruffini, M., 2022. Optical front/mid-haul with open access-edge server deployment framework for sliced O-RAN. IEEE Trans. Netw. Serv Manag. 19 (3), 3202–3219.

Moreira, C.M., Kaddoum, G., Baek, J.-Y., Selim, B., 2021a. Task allocation framework for software-defined fog v-ran. IEEE Internet Things J. 8 (18), 14187–14201.

Moreira, C.M., Kaddoum, G., Baek, J.Y., Selim, B., 2021b. Task allocation framework for software-defined fog v-RAN. IEEE Internet Things J. 8 (18), 14187–14201.

Muqing, W., Min, Z., 2019. Joint computation offloading and resource allocation in c-ran with mec based on spectrum efficiency. IEEE Access 7, 79056–79068.

Niknam, S., Roy, A., Dhillon, H.S., Singh, S., Banerji, R., Reed, J.H., Saxena, N., Yoon, S., 2020. Intelligent o-ran for beyond 5g and 6g wireless networks. arXiv preprint arXiv 2005, 08374.

O-ran: Towards an open and smart ran, ⟨https://www.o-ran.org/resources⟩ 2018.

Pamuklu, T., Mollahasani, S., Erol-Kantarci, M., 2021. Energy-efficient and delay-guaranteed joint resource allocation and DU selection in O-RAN. In 2021 IEEE 4th 5G World (October)Forum ((5GWF)) 99 104.

Pradhan, D., Priyanka, K.C., 2021. SDR network & network function virtualization for 5G Green communication (5G-GC). In Future Trends in 5G and 6G. CRC Press,, pp. 183–203.

Ranjbar, V., Girycki, A., Rahman, M.A., Pollin, S., Moonen, M., Vinogradov, E., 2022. Cell-free mMIMO support in the O-RAN architecture: a PHY layer perspective for 5G and beyond networks. IEEE Commun. Stand. Mag. 6 (1), 28–34.

Singh, S.K., Singh, R., Kumbhani, B., 2020. The evolution of radio access network towards open-ran: challenges and opportunities 2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), IEEE 1 6.

F.Z. Sun, Haijian and R.Q. Hu, “Joint offloading and computation energy efficiency maximization in a mobile edge computing system,” vol. 68, no. 3. IEEE, 2019, pp. 3052–3056.

L. Wang J. Zhou Y. Wang B. Lei Energy conserved computation offloading for O-RAN based IoT systems (May)In ICC 2022-IEEE International Conference on Communications 2022 4043 4048.