Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Non-orthogonal multiple access-based MEC for energy-efficient task offloading in e-commerce systems

Xiao Zheng[1,2†], Muhammad Tahir[3*†], Khursheed Aurangzeb[4], Muhammad Shahid Anwar[5*], Muhammad Aamir[6], Ahmad Farzan[7] and Rizwan Ullah[8†]

Abstract

Mobile edge computing (MEC) reduces the latency for end users to access applications deployed at the edge by offloading tasks to the edge. With the popularity of e-commerce and the expansion of business scale, server load continues to increase, and energy efficiency issues gradually become more prominent. Computation offloading has received widespread attention as a technology that effectively reduces server load. However, how to improve energy efficiency while ensuring computing requirements is an important challenge facing computation offloading. To solve this problem, using non-orthogonal multiple access (NOMA) to increase the efficiency of multi-access wireless transmission, MEC supporting NOMA is investigated in the research. Computing resources will be divided into separate sub-computing that will be handled via e-commerce terminals or transferred to edge sides by reutilizing radio resources, we put forward a Group Switching Matching Algorithm Based on Resource Unit Allocation (GSM-RUA) algorithm that is multi-dimensional. To this end, we first formulate this task allocation problem as a long-term stochastic optimization problem, which we then convert to three short-term deterministic sub-programming problems using Lyapunov optimization, namely, radio resource allocation in a large timescale, computation resource allocating and splitting in a small-time frame. Of the 3 short-term deterministic sub-programming problems, the first sub-programming problem can be remodeled into a 1 to n matching problem, which can be solved using the block-shift-matching-based radio resource allocation method. The latter two sub-programming problems are then transformed into two continuous convex problems by relaxation and then solved easily. We then use simulations to prove that our GSM-RUA algorithm is superior to the state-of-the-art resource management algorithms in terms of energy consumption, efficiency and complexity for e-commerce scenarios.

Keywords Mobile edge computing (MEC), E-commerce, MEC offloading, Energy optimization, Non-orthogonal multiple access (NOMA), Resource allocation

[†] Xiao Zheng, Muhammad Tahir and Rizwan Ullah contributed equally to this work.

*Correspondence:
Muhammad Tahir
muhammad.tahir.shaikh@gmail.com
Muhammad Shahid Anwar
shahidanwar786@gachon.ac.kr
Full list of author information is available at the end of the article

Introduction

With mobile edge computing (MEC) and the explosive growth of mobile Internet industries, the majority of shoppers are increasingly turning to the Internet for their shopping needs as e-commerce gains traction. Furthermore, expert e-commerce services are offered for every facet of transactions, hence cutting down on transaction expenses. As a result, an increasing number of conventional offline businesses are aggressively transforming into online businesses. Around 32.7 trillion yuan worth of transactions took place in China's e-commerce business in 2019. Even with e-commerce's enormous expansion perspective, there remain quite several important issues which have to be tackled. The difficulties associated with e-commerce are examined from three angles in this paper: system complexity, task-scheduling energy consumption, and data transmission energy consumption.

(1) Server load challenges to energy efficiency : I. High load leads to surge in energy consumption: E-commerce platforms usually need to handle a large number of user requests and data exchanges, especially during peak hours, when server load increases significantly. High load means that the server needs to invest more computing resources and power to maintain normal operation, resulting in a sharp increase in energy consumption. II. Inefficient energy utilization: During peak server load periods, due to unbalanced and unreasonable allocation of system resources, some servers may be overloaded while other servers are idle. In this case, energy utilization efficiency is low, resulting in unnecessary waste of energy.

(2) Challenges to energy efficiency caused by mobile device resource constraints: I. Limited battery life: The battery life of mobile devices is one of the key factors limiting their energy efficiency. E-commerce platform applications often require frequent interactions with the network, which can lead to rapid battery drain on mobile devices. In the case of limited battery life, users may not be able to use the e-commerce platform for a long time, thus affecting the user experience and the energy efficiency of the platform. II. Processing power and memory limitations: Mobile devices have relatively limited processing power and memory and cannot compare to servers. This results in mobile devices potentially experiencing delays when processing complex tasks or large amounts of data. To cope with this situation, e-commerce platforms may need to optimize their applications to reduce resource consumption, but this often affects the functionality and performance of the application.

To address these issues, this study presents mobile edge computing (MEC) to facilitate MEC-based e-commerce, where users can offload their tasks to neighboring edge servers [1–4]. Nevertheless, computing offloading in MEC systems is complicated and influenced by several variables [5, 6].

There are a few important factors to take into account when selecting an edge server for e-commerce: 1. Performance and processing capabilities: In order to handle the large amounts of data that e-commerce websites process, the high volume of concurrent access, and the demands of real-time trading, edge servers must be sufficiently performant and process capable. 2. Low latency and high availability: Because e-commerce websites must meet very strict real-time performance and availability standards, edge servers must have both of these qualities.

Regarding task scheduling, the MEC system needs to optimize task allocation to ensure that tasks are assigned to the most appropriate edge computing nodes for processing. This requires consideration of multiple factors such as node computing power, storage resources, network bandwidth, and task characteristics. Therefore, it is necessary to design an efficient task scheduling algorithm and formulate an optimal task allocation plan by analyzing the matching relationship between tasks and resources to achieve optimal utilization.

As the number of users in e-commerce scenarios increases, the resource collision problem becomes more and more serious. Non-orthogonal multiple access (NOMA) technology introduces interference information to achieve simultaneous transmission on the same frequency [7, 8], and uses serial interference removal technology for signal demodulation and interference elimination, effectively solving the resource collision problem caused by the increase in the number of users. However, there are still some challenging issues to be resolved. First, NOMA technology enables the simultaneous use of the same frequency and temporal resources by several users. This increases spectrum efficiency in e-commerce applications by enabling several users to transmit and interact with data simultaneously. This lessens the problems with resource conflicts brought on by an increase in user numbers. Secondly, NOMA technology achieves separation and correct demodulation of multi-user signals through power multiplexing and serial interference cancellation (SIC) technology. In e-commerce scenarios, when the number of users is large and resources are limited, interference may occur between signals, leading to resource conflicts. However, through the power multiplexing and SIC technology of NOMA technology, the signals of different users can be distinguished at the receiving end and multi-access interference is eliminated, thereby ensuring that each user can obtain stable communication quality.

The joint optimization of multi-dimensional resources in e-commerce based on NOMA mobile edge computing (NOMA-MEC) has attracted more and more research attempts. In the literature [9], Kiani et al proposed a computing resource offloading scheme that can implement user clustering based on NOMA and has the feature of low energy consumption. In the literature [10], the authors proposed the NOMA-based edge computing offloading problem and brought up a heuristic algorithm to reduce the energy required for computation by jointly optimizing power and time. However these initiatives only take short-term objectives into account, and efficiency degradation is expected when applying them to problems that require long-term optimizing.

Inspired by findings from the previously mentioned research, the article puts forward a method for multi-time-scale multi-dimensional resource allocation for NOMA-MEC for e-commerce platforms. The paper aims to optimize resource unit allocation and task decomposition simultaneously to minimize continuous utilization of all e-commerce platform devices depending on long-term queue delay restrictions. As a result, this work first divides the long-term stochastic joint optimization challenge into three short-term deterministic tasks (i.e., task computation, task partitioning, and wireless spectrum allocation) applying the Lyapunov method [11].

E-commerce devices and resource units are particularly grouped by employing clustering-based methods to reduce complexity. The allocation of wireless spectrum is characterized as a large-scale, one-to-many matching process that is dealt with at the base station (BS) level. Resource unit allocation should be implemented via group swap matching. Switching and matching occur inside each group after the devices and resource units for the e-commerce platform have been separated into multiple groups. Subsequently, tasks are divided, and computer resources are distributed and assigned in shorter time intervals on the device side. The contributions of the work are as follows:

(1) Optimization decomposition of multidimensional problems: Three feasible deterministic problems are derived from the long-term stochastic multidimensional optimization task employing Lyapunov optimization.
(2) Group switch matching for resource allocation: The group swap matching-based resource allocation techniques offers a practical, straightforward, and flexible solution to the interdependency problem between various resources and e-commerce terminals.

The organization structure of this paper is as follows: Introduction section is an introduction, Related work section is related work, System model section is the system model, Problem description and analysis section is problem description and analysis, task division and resource allocation are in Partitioning tasks and allocating resources section, and analysis of experimental results is in Simulation results section, Conclusion section is the conclusion, Funding is the acknowledgments.

Related work

MEC provides cloud and IT-related services at the radio access network (RAN) near mobile users (MUs). [12, 13]. To supply contextually aware services and services that offer distinctive mobile browsing experiences, app designers and content vendors can leverage the RAN edge, which offers an ultra-low latency and high-bandwidth service setting. Additionally, applications have instant access to real-time wireless data from the network (such as location-based data, cell loads, etc.). By enabling resource organization [14, 15] and service architecture [16], MEC enhances edge response by accelerating content, services, and apps. Thus, by running networks and services more efficiently, the user experience can be enhanced.

Edge computing is gaining popularity as a complement to, and expansion of cloud computing [17]. By adopting a distributed computing strategy, edge computing eliminates the need for devices to upload data to cloud servers and server power consumption, and enhances security, and latency. Users' computing duties are divided across several servers located throughout the network [18]. This method tries to solve the problem of network congestion and significant transmission delay brought on by cloud computing's centralized computation. Additionally, the real-time performance of the data calculating process is further ensured by the edge server's ability to respond to the user's request and task in a shorter amount of time [19]. At order to reduce the large data transmission delay experienced during long-distance communication, the server is placed in a network edge node that is closer to the device.

The following studies include the use of edge computing in a consumer IoT environment. To achieve the optimal distribution of widely distributed green and energy-saving computer resources, time and energy costs are optimized in the literature [20]. The offloading issue of multi-hop computing jobs in a hybrid edge cloud computing environment is researched in the literature [21], and the offloading technique that satisfies the service quality requirements is accomplished through the game method. A hybrid computing structure with intelligent resource planning is suggested in the literature [17] to meet real-time needs. In conclusion, edge computing, which is installed near e-commerce devices at the network's edge, offers suitable computing resources for these devices, which can lower system expenses and satisfy task service quality requirements in a range of situations.

In short, the resource limitations of mobile devices such as processor performance, memory size, and battery capacity prompt e-commerce application developers to pay more attention to resource optimization. By optimizing strategies, reducing unnecessary resource consumption and improving resource efficiency, developers can provide users with a smoother and more responsive application experience, thereby improving user satisfaction.

System model

The basic idea of NOMA is to allocate non-orthogonal communication resources to different users at the transmitting end. In the orthogonal scheme, if a piece of resource is evenly allocated to N users, then subject to the constraints of orthogonality, each user can only be allocated $\frac{1}{N}$ resources. NOMA gets rid of the limitation of orthogonality, so the resources allocated to each user can be greater than $\frac{1}{N}$. In the extreme case, each user can be allocated to all resources to realize resource sharing among multiple users. The conventional NOMA technology is based on the condition that the user's CSI is known and correct when conducting theoretical analysis. This work is carried out based on the above conditions.

Consider concerning the NOMA and MEC combined e-commerce platform situation shown in Fig. 1, which consists of N access points (APs) and a base station (BS). Single-antenna architecture has been widely adopted in various MEC networks. However, the optimization scenarios proposes in the work can be extended to scenarios involving multiple antenna ensembles. The wireless channel connects each AP in the system to the base station, which is regarded as the user m of the AP. It provides a set of single-antenna services and provides wireless access and computing services to $\mathcal{M}$ e-commerce terminal devices. The system model is represented by M nodes: $\mathcal{M} = \{1, 2, ..., m, ...M...\}$. The terminal tasks under the e-commerce platform are either partially offloaded or offloaded to a nearby BS for processing.

In contrast to [22], this work considers a discrete time slot structure in which the optimization method is divided into $\mathcal{K}$ time slots, with a κ duration for each time slot. The formula $\mathcal{K} = \left\{1, ..., k, ..., K\right\}$ indicates a sequence of time slots. Investigate a case that is nearly static, in which the CSI changes from slot to slot but doesn't change within a slot. $t \in T, t = \{1, ..., T\}$ denotes a time epoch that is associated with each successive K' slot.

The expression $\mathcal{K}(t) = \left\{(t-1)K' + 1, ..., tK', tK' + 1, ...(t+1)T'\right\}$ specifies the tth time epoch. $\mathcal{P} = \{1, ..., p, ..., P\}$ is the collection of the resources, which are split into P time-frequency resource components, B bandwidth, and K' time period. Challenges of splitting and multi-dimensional resource distribution are examined. On the primary time scale, the unlimited resource allocation strategy is optimized at the start of every time period. The binary indicator $\boldsymbol{I(t)} = \left\{I_m^p(t), m \in \mathcal{M}, p \in \mathcal{P}\right\}$, where $I_m^p(t) = 1$, reflects the resource allocation strategy. In the t-th time interval, $I_m^p(t) = 1$ indicates that resource unit p is assigned to device m; else, $I_m^p(t) = 0$. Subsequently, combined optimization computation of resource allocation and splitting is carried out on a small time scale, depending on the resource unit's allocation technique used for each time slot.

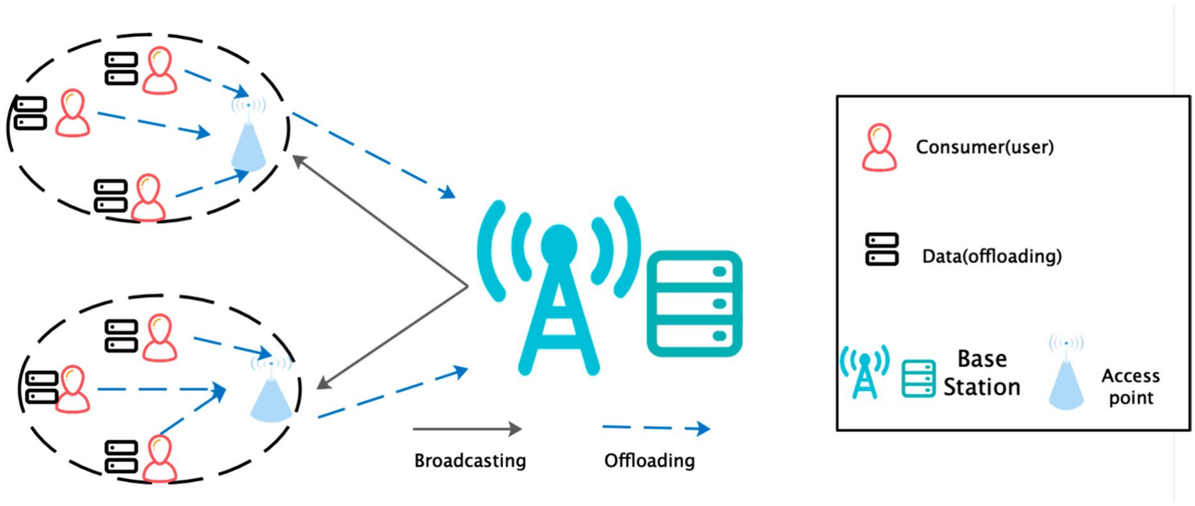

Fig. 1 Typical NOMA-MEC for e-commerce platform

First, the time epoch provides a globally consistent time base so that various parts of the system can be synchronized more accurately. This is especially important for real-time processing systems that require a high degree of coordination. Second, in systems that require dynamic allocation of resources, time epochs can help manage resources more efficiently.

A. Traffic model on the device A task splitting approach is used in the paper [23] that permits the division of each task into independent subtasks of size X_0 (bits). W orthogonal channels are shared by W users who are active without any co-channel interruption. It is assumed that during the kth time slot, the $a_m^{max}(t)$ subtask reaches device m. The division of arriving tasks into two distinct and parallel types can be done: for local operations, $a_m(k)$ subtasks and for task offloading, $a_m^{max}(k) - a_m(k)$ subtasks. Taking everything into consideration, the task to be split at device m during time slot k is specified as

$$\begin{cases} X_m^L(k) + X_m^O(k) = a_m^{max}(t)X_0, \\ X_m^L(k) = a_m(k)X_0, a_m(k) \in \{0, 1, ..., a_m^{max}(k)\}. \end{cases} \quad (1)$$

where $X_m^L(k)$ denotes the length of the task that device m is currently processing locally. The task length when device m performs compute offloading at time k is denoted by $X_m^O(k)$. The local processing and resource offloading are executed using $Q_m^L(k)$ and $Q_m^O(k)$, respectively. $Q_m^L(k)$ and $Q_m^O(k)$ [24] vary as you can observe in Fig. 1.

$$Q_m^L(k+1) = max\{Q_m^L(k) - d_M^L(k), 0\} + X_m^L(k), \quad (2)$$

$$Q_m^O(k+1) = max\left\{Q_m^O(k) - d_M^O(k), 0\right\} + X_m^O(k) \quad (3)$$

where $d_M^L(k)$ and $d_M^O(k)$ respectively indicate the volume of data departure $Q_m^L(k)$ and $Q_m^O(k)$.

B. Locally task data processing scheme The kth time slot's local processing amount of data is specified as

$$d_M^L(k) = \kappa\frac{f_m(k)}{l_m}, \quad (4)$$

where the number of CPU cycles assigned to device M within the kth time slot is denoted by $f_m(k)$. l_m stands for the amount of processing, or the number of CPU cycles needed for each bit. The local operating computing delay and associated energy consumption produced by the device m in the kth time slot are specified as

$$D_m^L(k) = min\{\tau, Q_m^L(k)l_m/f_m(k)\} \quad (5)$$

$$E_m^L(k) = \iota_m f_m^3(k) min\{\kappa, Q_m^L(k)l_m/f_m(k)\} \quad (6)$$

where the chip structure regulates a constant power coefficient named τ_m.

C. Computing offload processing method interference cancellation (SIC). The BS correctly sequences the decoding of signals from devices which have greater channel gains, but all other communications are regarded as interference. $h_m^p(k)$ is the uplink channel gain connecting device m to resource unit p within the kth time slot. The following is the signal-to-noise ratio (SNR) data that the BS acquired:

$$SNR_m^p(k) = \frac{g|h_m^p(k)|^2}{\sum_{i=1, i\neq m}^{M}[y_i^p(t)g|h_i^p(k)|^2] + (\theta)^2} \quad (7)$$

The power of transmission is g. The first component of the denominator is the additive white Gaussian noise power, yet the second item is intra-cell interference from other lower channel gain devices.

Note: First, the base station decodes the signal strategy, that is, the base station decodes the signal in order of channel gain. This is an optimization strategy designed to improve decoding efficiency and accuracy, especially in the presence of multiple signals with interference between them. Second, it was mentioned that users seem to treat all signals in Eq. 7 as interference. This reflects the way users actually handle signal processing, that is, users may not adopt a decoding strategy like the base station, but instead treat all signals as potential interference.

Consequently, the following formulas are used to compute the data amount of the task that can be offloaded in the kth time slot and the device m's transmission speed applying the resource unit p.

$$R_m^p(k) = Blog_2[1 + SNR_m^p(k)], \quad (8)$$

$$d_m^O(k) = \kappa\sum_{p=1}^{P} I_m^p(s)R_m^p(k). \quad (9)$$

First, in edge computing networks, resources are limited, and by arranging users in descending order, the system can more easily identify and handle those users that have the greatest impact on performance. This descending order helps ensure resources are allocated to users who need them most, optimizing overall network performance. Secondly, by giving better services to users with higher priority (such as higher data transmission rates, lower delays, etc.), the service quality of these users can be improved.

All mobile user devices are arranged in decreasing order in accordance with the $\varepsilon_i^{p_*}|_{\sigma(p_i)=m_s}$, $\forall m_s \in \mathcal{M}$ criterion to produce the preference set $\xi(p_i) = \{..., m_s, m_p, ...\}$. When all B2B pairs are available, they are sorted according to $\varepsilon_s^{m_*}|_{\sigma(m_s)=p_i}$ to produce $\xi(m_s)$, which stands for the preference profile of m_s. Create

$\Xi = \{\xi(p_1), ...\xi(p_m), \xi(m_1), ..., \xi(m_S)\}$ to represent the entire set of preferences.

Here, arranging $\varepsilon_s^{m_*}$ in descending order means that the services with the highest matching degree are placed first, so that those services with the highest matching degree can be prioritized to improve user experience. Generating a preference set p_i means that it is convenient for users to make decisions, so that they can find services that meet their needs more quickly and reduce selection costs.

The following indicates the corresponding energy consumption that device m produced during the kth time slot:

$$E_m^O(k) = pmin\left\{\tau, \frac{Q_m^O(k)}{\sum_{p=1}^{P} I_m^p(t) R_m^p(k)}\right\}. \tag{10}$$

$$\varepsilon_i^{p_*} = \Gamma_i^p(\xi_i^p)|_{\sigma(p_i)=m_s} = \frac{\Gamma_i^p(\xi_i^{p_*})}{E_i^p(\xi_i^{p_*})} \tag{11}$$

The composition of the modeling given above is the construction of traffic on the device, local task processing and task offloading scheme, and various mathematical symbols and formulas to be used are also given.

Problem description and analysis

First, queuing delay restrictions are introduced in this section. This is followed by the idea put forward of the multi-dimensional resource allocation and task splitting optimization problem.

A. Queued delay constraint To guarantee the efficiency and promptness of task offloading, manage queuing delay restrictions. According to Little's law [25], the following formula is used to determine the queuing delays of $Q_m^L(k)$ and $Q_m^O(k)$.

$$\lim_{T\to\infty} \frac{1}{K} \sum_{k=1}^{K} \frac{Q_m^L(k)}{MA_m^L(k)} \le D_{m,max}^L \tag{12}$$

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \frac{Q_m^O(k)}{MA_m^O(K)} \le D_{m,max}^O \tag{13}$$

where $MA_m^L(k)$ and $MA_m^O(k)$ are the average data arrival rate of moving time of $Q_m^L(k)$ and $Q_m^O(k)$, respectively. $D_{m,max}^L$ and $D_{m,max}^O$ are their respective maximum tolerable queuing delays.

B. Problem definition The aim is to minimize the total accumulated long-term energy consumption of all devices, according to queue latency restrictions, by centrally optimizing resource unit allocation, partitioning, and computing task scheduling.

$$\boldsymbol{P1} : \underset{\boldsymbol{I,a,f}}{max} \overline{E} =$$
$$\frac{1}{K} \sum_{k=1}^{K} \sum_{m=1}^{M} E\left\{E_m^L(k) + E_m^O(k) + \Gamma_s^M(\xi_s^m) + \Gamma_i^P(\xi_i^p)\right\}$$
$$\text{s.t. } C_1 : a_m(k) \in \left\{0, 1, ..., a_m^{max}(k)\right\}, \forall m \in \mathcal{M}, \forall k \in \mathcal{K},$$
$$C_2 : 0 \le f_m(k) \le f_m^{max}, \forall m \in \mathcal{M}, \forall k \in \mathcal{K},$$
$$C_3 : I_m^P(t) \in 0, 1, \forall m \in \mathcal{M}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T},$$
$$C_4 : \sum_{p=1}^{P} I_m^p(t) \le 1, \forall m \in \mathcal{M}, \forall t \in \mathcal{T},$$
$$C_5 : \sum_{m=1}^{M} I_m^p(t) \le M_p, \forall p \in \mathcal{P}, \forall t \in \mathcal{T},$$
$$C_6 : I_m^p(t) SNR_m^p(k) \ge SNR_m, \forall m \in \mathcal{M}, \forall k \in \mathcal{K},$$
$$C_7 : 0 \le g_s^m \le max(g_s^m),$$
$$C_8 : 0 \le g_i^p \le max(g_i^p).$$

The resource group allocation vector is denoted by $\boldsymbol{I} = \{\boldsymbol{I}(\boldsymbol{t})\}, t \in \mathcal{T}$, the wireless dividing of resources vector is expressed by $\boldsymbol{a} = \{\boldsymbol{a}(\boldsymbol{k})\}, k \in \mathcal{K}$. The vector for assigning computing resources is indicated by $\boldsymbol{a}(\boldsymbol{k}) = \left\{a_m(k), m \in \mathcal{M}\right\}$, and $\boldsymbol{f} = \left\{\boldsymbol{f}(\boldsymbol{k}) = f_m(k), m \in \mathcal{M}, k \in \mathcal{K}\right\}$. The restriction on resource dividing is C_1. C_2 is an expression of the device's computing resource allocations constraint. Each device is allowed to utilize a maximum of one resource entity, denoted by $C_3 - C_5$. At most M_p devices can acquire the resource group p. The resource groups assigned to device m must be achieved by C_6 to guarantee that the SNR acquired at the BS exceeds the minimal allowed SNR_m. C_7 ensures that m_s's power allocation does not go beyond the maximum permitted transmission power $max(g_s^m)$, C_8 ensures that p_i's power allocation does not go beyond the maximum permitted transmission power $max(g_i^p)$.

C. Transformation of the problem It is challenging to find a direct solution for P1, a non-deterministic polynomial (NP) hard situation.

Lyapunov optimization is usually applied to dynamic systems such as resource allocation problems and scheduling problems with stability and performance optimization requirements. It can be used to solve practical problems such as how to maximize network throughput, minimize user average delay, and minimize total network power consumption. Since the Lyapunov function can effectively handle system uncertainty, it can show good robustness to a certain extent.

In the model, multiple short-term determinate subproblems are derived from the original long-term stochastic optimization problem-solving through the application of Lyapunov optimization [26, 27]. The queue stability restriction condition of the virtual queue theory can be determined by Eqs. (11) and (12). $\Delta_m^L(k)$ and $\Delta_m^O(k)$ are matching virtual queues that can be transformed into

$$\Delta_m^L(k+1) = max\left\{VQ_m^L(k) + \frac{Q_m^L(k)}{MA_m^L(k)} - D_{m,max}^L, 0\right\} \quad (14)$$

$$\Delta_m^O(k+1) = max\left\{VQ_m^O(k) + \frac{Q_m^O(k)}{MA_m^O(k)} - D_{m,max}^O, 0\right\} \quad (15)$$

Formula (11) and (12) are inevitably preserved once the average rates of $\Delta_m^L(k)$ and $\Delta_m^O(k)$ remain stable. Accordingly, problem P_1 will be converted into

$$\begin{aligned} \boldsymbol{P2}: \min_{\boldsymbol{I(t),a(k),f(k)}} \sum_{m=1}^{M} V\iota_m f_m^3(k)\min[\kappa, \frac{Q_m^L(k)l_m}{f_m(k)}] + \\ Vg\min[\kappa, \frac{Q_m^O(k)}{\sum_{p=1}^{P} I_m^p(t)R_m^p(k)}] + \\ Q_m^L(k)[a_m(k)X_0 - \kappa\frac{f_m(k)}{l_m}] + \\ Q_m^O(k)[(a_m^{max}(k) - a_m(k))X_0 - \kappa\sum_{p=1}^{P} I_m^p(t)R_m^p(k)] + \\ \Delta_m^L(k)[\frac{Q_m^L(k)}{\frac{1}{k}[\sum_{i=1}^{k-1} X_m^L(i) + a_m(k)X0]}] + \\ \Delta_m^O(k)[\frac{Q_m^O(k)}{\frac{1}{k}[\sum_{i=1}^{k-1} X_m^O(i) + (a_m^{max}(k) - a_m(k))X_0]}] + \\ \Gamma_s^M(g_s^m) + \Gamma_i^M(g_i^p) \end{aligned} \quad (16)$$

s.t. C_1 - C_8

We may deduce that $P2$ will be split up into three subtasks of optimization: $SP1$, which is the sub-problem of resource group allocation; $SP2$, which is the sub-problem of task splitting; and $SP3$, which is the sub-problem of computing the allocation of resources.

Partitioning tasks and allocating resources

This section provides an overview of the claimed schemes before introducing the three deconstructed subschemes and the corresponding reactions.

A. Resource unit allocation optimization Device m along with BS decide a resource group allocation technique at the start with every interval in $SP1$. Since the values of $h_m^p(k)$, $Q_m^O(k)$, and $R_m^p(k)$ differ over the time slot, their experience $\overline{h_m^p(k)}$ is the mean value.There are two considerations: $\overline{Q_m^O(k)}$ and $\overline{R_m^p(k)}$. Consequently, $SP1$ will refer to:

$$\begin{aligned} \boldsymbol{SP1}: \min_{\boldsymbol{I(t)}} \sum_{m=1}^{M} Vg\min[\kappa, \frac{\overline{Q_m^O(t)}}{\sum_{p=1}^{P} I_m^p(t)\overline{R_m^p(k)}}] - \\ \overline{Q_m^O(t)}\tau\sum_{p=1}^{P} I_m^p(t)\overline{R_m^p(t)} - \\ (\Gamma_s^M(g_s^m) + \Gamma_i^P(g_i^p)) \end{aligned} \quad (17)$$

s.t. C_3 - C_8

The typical approach for $SP1$ is a one-to-many matching between devices and resource groups. The following is the performance of the one-to-many matching algorithm η that we symbolizes here:

$$\begin{aligned} &(1)\upsilon(m) \subset \mathcal{P} \cup \{\emptyset\}, and|\upsilon(m)| \in \{0,1\}, \forall m \in \mathcal{M}, \\ &(2)\upsilon(p) \subset \mathcal{M}, and|\upsilon(m)| \le M_p, \forall p \in \mathcal{P}, \\ &(3)\upsilon(m) = p \Leftrightarrow m \in \lambda(p), \forall p \in \mathcal{P}, \forall m \in \mathcal{M}. \end{aligned} \quad (18)$$

The restrictions C_4 and C_5 are discussed in the scenarios (1) and (2) above, whereas the third situation demonstrates that resource unit p is allocated to device m and vice versa if device p and resource unit p match. Specifically, there is a close relationship between υ and the resource unit allocation rating $I_m^p(t)$.

$$\begin{cases} I_m^p(t) = 1, \text{if} \quad \upsilon(m) = p, \\ I_m^p(t) = 0, \quad \text{otherwise}. \end{cases} \quad (19)$$

The following are the utility functions for device m and resource unit p:

$$\begin{aligned} U_m(p) = -Vg\min[\kappa, \frac{\overline{Q_m^O(t)}}{\sum_{p=1}^{P} I_m^p(t)\overline{R_m^p(k)}}] + \\ |\overline{Q_m^O(t)}\tau I_m^p(t)\overline{R_m^p(t)}| + \\ \Gamma_s^M(g_s^m) + \Gamma_i^P(g_i^p) \end{aligned} \quad (20)$$

$$\begin{aligned} U_p(m) = \sum_{m=1}^{M} |\{-Vg\min[\kappa, \frac{\overline{Q_m^O(t)}}{I_m^p(t)\overline{R_m^p(t)}}]| + \\ |\overline{Q_m^O(t)}\tau I_m^p(t)\overline{R_m^p(t)}|\} + \\ \Gamma_s^M(g_s^m) + \Gamma_i^P(g_i^p) \end{aligned} \quad (21)$$

It is not feasible to match every device and resource unit as a result of the high matching complexity of a large-scale e-commerce network. Consequently, we start by grouping the resource units and devices into collections. Devices and resource groups are specifically classified into Y sets, i.e., $\mathcal{M} = \{\mathcal{MG}_1, ..., \mathcal{MG}_y, .., \mathcal{MG}_Y\}$, in accordance with the clustering technique [28].The expressions $\mathcal{P} = \{\mathcal{PG}_1\}$ and $\mathcal{PG}_y$. Devices and resource groups ($M_g = M/y$ and $P_g = P/y$, respectively) are available to each group. Regulated resource units, PG_Y, are resources that are commonly assigned to MG_Y for offloading by devices inside each union. After then, switch matching is carried out in a semi-distributed fashion in each group in order to prioritize devices and resource units based on their utility, ranking them from high to low. By group switching and matching at the two points, the resource unit allocation constraint is thereby resolved.

Definition 1 Matching υ and two device resource group pairings $(m, p), (v, l) \in \upsilon$ are defined as follows: $\upsilon(p) = m$ and $\upsilon(v) = l$, $\forall m \neq v$ and $m, v \in \mathcal{MG}_s$, $\forall p \neq l$ and $p, l \in \mathcal{PG}_s$, if they satisfy

$$\begin{aligned} U_m(l) \leq U_m(p) \quad and \quad U_v(m) \leq U_v(l), \\ U_p(v) > U_p(m) \quad and \quad U_l(m) > U_p(v) \end{aligned} \tag{22}$$

The equation $\upsilon_{mv}^{pl} = \{\upsilon \setminus (m,p),(v,l))\} \cup \{(m,l),(v,p)\}$ illustrates the method of exchanging variables υ and $\upsilon_{mv}^{pl} \succ \upsilon$.

Definition 2 The matched υ is bilaterally interchange-stable if there isn't a swap match.

The details of the group swap matching-based resource unit allocation technique are compiled in Algorithm 1. Devices and resource groups are assigned to Y unions during initialization, and each device unity is given a collection of resource units, denoted as $\mathcal{MG}_y \leftarrow \mathcal{PG}_y$. Next, the allocation group's devices and resource groups will be paired at arbitrary, provided that all of the conditions in Eq. (15) are satisfied. Formula (20) determines the preferences generated by every device and resource group.

When transfer matching occurs, the program moves to its preferred position in PG_y , the resource group l for every device m in MG_y that presently matches resource group p in PG_y. The new matching υ_{mv}^{pl} replaces the old matching υ for each current device v in MG_y that matches l in PG_y if and only if $\upsilon_{mv}^{pl} \succ \upsilon$ and satisfies (15). In every other scenario, υ keeps the same. Till no matches are swapped, the process will terminate.

To determine the resource unit allocation indicator $I^*(t)$, the ultimate υ is transformed into it using formula (19).

Algorithm 1 Group Switching Matching Algorithm Based on Resource Unit Allocation (GSM-RUA)

```
Input: $\mathcal{M}, \mathcal{P}, \overline{h_m^p(t)}, \overline{Q_m^O(t)}, V, \kappa, SNR_m, M_p, g$
Output: $\boldsymbol{I^*(t)}$.
Step 1: Initialization
Resource groups and devices are divided into $Y$ pairs, and $PG_y$ is paired with
$MG_y$ to facilitate task offloading.
The resource groups in $PG_y$ and the devices in $MG_y$ will be matched at arbitrary,
provided that all the requirements in (15) are satisfied. Every resource group and
device takes (21) into account while constructing the preference list.
Step 2: Switching Matching
while there has a switch matching do
    Set the existing match, $\upsilon$, each device $n$ in $MG_y$ that currently matches the
    resource unit $p$ in $MG_y$ takes a program to its optimal resource unit $l$ in $MG_y$.
    for every device $v$ in $MG_y$ that is currently matched with in $l$ in $PG_y$ do
        if $\upsilon_{mv}^{pl} \succ \upsilon$ and all constrains in (15) are satisfied
        then
        $\upsilon_{mv}^{pl} \rightarrow \upsilon$
        else
        $\upsilon$ remains same.
        end if
    end for
end while
Step 3: Resource Unit Allocation
The final $\upsilon$ is converted to the resource unit allocation indicator $\boldsymbol{I^*(t)}$ based on
(19).
```

B. Task splitting and resource allocation optimization The following factors are crucial for the joint optimization of task splitting and computing resource allocation in the context of e-commerce: (1) High task calculation volume, high data processing volume, and high complexity tasks necessitate more splitting and refinement; (2) For tasks requiring quick response, splitting should guarantee that each sub-task can be finished in a shorter amount of time.

The following is the formulation of the task splitting challenge. The task splitting decision-making between local operations and offloading is assigned by *SP*2 in the *k*th time slot.

$$\begin{aligned} \boldsymbol{SP2}: \min_{a_m(k)} \Gamma(a_m(k)) = \\ |Q_m^L(k)a_p(k)X_0 + Q_m^O(k)[(a_m^{max}(k) - a_m(k))X_0]| + \\ |\Delta_m^L(k)[\frac{Q_m^L(tk)}{\frac{1}{k}[\sum_{i=1}^{k-1} X_m^L(i) + a_m(k)X_0]}]| + \\ |\Delta_m^O(k)[\frac{Q_m^O(k)}{\frac{1}{k}[\sum_{i=1}^{k-1} X_m^O(i) + (a_m^{max}(k) - a_m(k))X_0]}]| + \\ \mathit{\Gamma}_y^M(g_y^m) + \mathit{\Gamma}_i^P(g_i^p) \end{aligned} \tag{23}$$

s.t. C_1

The amount of CPU cycle frequencies that each device assigns for local processing in the *k*th time slot is managed by the computing allocation of resources subscheme *SP*3, and this quantity is determined by the following procedure.

$$\begin{aligned} \boldsymbol{SP3}: \min_{\boldsymbol{f(k)}} = \\ |V\iota_m f_m^3(k) min[\kappa, \frac{Q_m^L(k)l_m}{f_m(k)}]| - \\ |Q_m^L(k)\tau\frac{f_m(k)}{l_m}Q_m^L(k)| \end{aligned} \tag{24}$$

s.t. C_2

Lagrangian dual decomposition provides an easy way to address convex optimization challenges such as SP2 and SP3.

According to the above conditions, the original constrained problem is transformed into an unconstrained problem through the Lagrangian function. If the original problem is difficult to solve, the dual problem is used to replace the original problem under the condition of satisfying KKT, which makes the problem solving easier.

This study introduces a one-to-one matching model to address problem P_1. This model matches users in accordance with their shared preferences and block-to-block pairs (B2B). This allows the original NP-hard problem to be split into two distinct subproblems and addressed in a straightforward manner. In this study, the created matching problem is represented by the triplets $(\mathcal{N}, \mathcal{M}, \mathcal{P})$, where $\mathcal{P}$ is the collection of shared preferences and $\mathcal{M}$ and $\mathcal{N}$ are two finite and distinct sets of B2B couples and users, respectively. In order to improve energy efficiency, both B2B pairs and individual devices work to establish an appropriate channel reuse cooperative relationship within the confines of QoS and transmit power.

Definition 3 In the case of a match $(\mathcal{N}, \mathcal{M}, \mathcal{P})$, σ is represented as a pointwise mapping from $(\mathcal{N}, \mathcal{M}, \mathcal{P})$ to itself. In other words, $\forall n_k \in \mathcal{N}$ and $\forall m_i \in \mathcal{M}$, $\sigma(n_k) \in \mathcal{M} \cup \{n_k\}$ and $\sigma(m_i) \in \mathcal{N} \cup \{m_i\}$. $\sigma(n_k) = m_i$ iff $\sigma(m_i) = n_k$.

In the case of $\sigma(m_i) = m_i$ or $\sigma(n_k) = n_k$, then m_i or n_k maintains a single value. According to their preferences, either m_i or n_k can send a request to create a partnership with their selected partner and then show how the assigned transmission power for the resulting partnership (i.e., the power allocation subproblem) works. Both m_i and n_k make the assumption that they are just interested in their own pairings and are not very interested in the pairings of others.

The resource allocation technology based on group switch matching algorithm has good scalability in e-commerce scenarios, specifically: First, the resource allocation technology of group switch matching achieves flexible allocation of resources by dividing resources into different groups and matching and switching between these groups. This technology can adapt to the increase in the number of users by increasing the number of groups or adjusting the size of the groups. At the same time, as the number of computing tasks increases, this technology can ensure efficient resource utilization and delayed response by optimizing the matching algorithm and exchange mechanism.

C. Stable and energy-efficient matching In order to match B2B pairs and user equipments, the study proposed Algorithm 2, which employs the GS method after obtaining $P(m_i)$ and $P(n_k)$, $\forall m_i \in \mathcal{M}$, $\forall n_k \in \mathcal{N}$ [29, 30]. In the initial iteration, each $m_i \in \mathcal{M}$ sends a collaboration request to its most preferred user equipment $max\left\{\varepsilon_i^{m_*}|_{\sigma(m_i)=n_k, \forall n_k \in \mathcal{N}}\right\}$. The request is then received by each $n_k \in \mathcal{N}$, which, if it has a superior candidate, rejects the B2B pair. If $m_i \in \mathcal{M}$ accepted as a candidate at this point has not been rejected by the user. The user with the highest priority in the set of users who have not yet issued a refusal receives a fresh request from $\forall m_i \in \mathcal{M}$ that has been refused in the following phase. When a B2B pair receives rejections from all of its preferred users, it gives up and stops sending requests. Only the most favored B2B pair is accepted for $\forall n_k \in \mathcal{N}$ after all incoming requests, including candidate requests kept from earlier steps, have been compared. When $\forall m_i \in \mathcal{M}$ has found a mate or has had all of its requests denied by users, the request sending and rejection procedure comes to a conclusion. The best candidate preserved at any step may subsequently be eliminated if a better candidate appears, which is a feature of Algorithm 2.

Algorithm 2 Energy-effective stable matching algorithm

```
**Input**: $\mathcal{N}$, $\mathcal{M}$, $\Xi$
**Output**: $\sigma$.
**Initialize**: $\sigma = \omega$, $\Omega = \mathcal{M}$
**while** $\Omega \neq \omega$ **do**
    **for** $m_i \in \Omega$ **do**
        $m_i$ chooses the user with the highest ranking from $\Xi(m_i)$.
    **end for**
    **for** $n_k \in \mathcal{N}$ **do**
        **if** $n_k$ accepts a request from $m_i$ and chooses $m_i$ instead of the present candidate $m_j$ received in the previous phase, $m_i \succ n_k$, $m_j$ **then**
        $m_i$ is taken into account as a new candidate while $m_j$ is rejected by $n_k$, i.e. $\sigma(n_k) = m_i$;
        Remove $n_k$ from $\Xi(m_i)$, take away $m_i$ from $\Omega$, and add $m_j$ to $\Omega$.
        **else**
        $m_i$ is rejected by $n_k$, which means $m_j$ is retained as its candidate, proving that $\sigma(n_k) = m_j$.
    Deleting $n_k$ from $\Xi(m_i)$.
        **end if**
    **end for**
**end while**
```

D. Our proposed algorithm The three steps in the put forward algorithm are as follows:

> First: Define all resource unit allocation rules and queue backlog indicators to zero.
>
> Second: In accordance with Algorithm 1, each device partially distributes the optimal resource group allocation $\boldsymbol{I^*(t)}$ and employs the amount of resource unit that has been assigned to transmit data.
>
> Third: Every device gathers up the best techniques for allocating tasks and resources. Power consumption, queue overflow, queue latency, and updates for $Q_m^L(k+1)$, $Q_m^O(k+1)$, $VQ_m^L(k+1)$, and $VQ_m^O(k+1)$ will all be taken into consideration for each device. These factors will then be taken into account using the formulas (2), (3), (13) and (14) . The iteration continues between the second and third phase until $k > K$.

The optimization decomposition approach may break down the complicated energy consumption problem into a number of smaller issues, allowing it to optimize each smaller issue in terms of energy consumption. In addition, the network topology is optimized. For example, edge computing technology is used to integrate the data centers of various e-commerce platforms, therefore reducing the quantity and distance of data transmission, which can also efficiently cut energy consumption.

Therefore, the optimization decomposition method significantly lowers energy usage and raises the caliber of services provided by e-commerce platforms. It makes it possible to optimize and enhance the platform more precisely, which lowers energy consumption and raises service quality, by breaking down complicated difficulties into a number of smaller problems.

Simulation results

The following part assesses the proposed method using simulations. The investigation looked at a single cell that has a radius of 2000 US dollars millimetres. 20 devices make up each of the 100 groups; this division of resource units and devices is consistent.Specific simulation parameters are listed in Table 1.

We contrast two cutting-edge algorithms. The first is the resource unit allocation algorithm (SMRA) inspired by switch matching that was put forth by [16], which the basic idea is to study the energy efficiency of an uplink hybrid system integrating NOMA into OMA (HMA) to support a large number of e-commerce devices. It is significant to notice that SMRA energy efficiency maximization is replaced in the scenarios with energy consumption minimization. The creation of the Access Control and Resource Allocation Algorithm (ACRA) is the cause of the second. ACRA relies on pricing matching and Lyapunov optimization, and in order to determine the optimal decision, it needs a perfect GSI. In this case, still it presumes that only the CSI of the prior slot is available, meaning that the CSI contains out-of-date information. The local computer resources are configured to the largest value possible between SMRA and ACRA, and the task partitioning step is assigned arbitrarily.

Discuss the computational complexity of the work. For the initial stage, $O(n^2M)$ operations are required. In the exchange-matching stage, it means that the number of iterations to reach the final match is I_1. In each iteration, all possible exchange combinations should be considered, which requires $O(n^2)$ operations. Indicates that the computational complexity of devices for computing energy efficiency is $O(x)$, and then, the total complexity of the swap-matching stage is $O(I_1 n^2 X)$.

In order to achieve the minimal energy consumption and minimum delay requirements, the GSM-RUA algorithm in this article is designed with the following measurement parameters in mind: (i) average handover delay (i.e., the average time of all successful handovers). (ii) The total energy used during the simulation process as well as the average energy used during the switching procedure.

Table 1 Simulation parameters

Parameters	Values	Parameters	Values
K	200	M	400
P	2000	M_p	8
l_m	1000cycle/b	κ	2s
δ^2	-118dBm	ι_m	2×10^{-30}Watt $\cdot$ $s^3/cycle^3$
B	0.20 MHz	g	25dBm
X_0	10^5 bits	f_m^{max}	2×10^8 cycle/s
$D_{m,max}^L$	5s	$D_{m,max}^O$	5s
K_0	15	SNR_m	5dB
$a_m^{max}(k)$	[30,40]	V	8

The average energy consumption of edge devices as an index of time slots is illustrated in Fig. 2. Both SMRA and ACRA have a task splitting percentage of 0.85, meaning that 85% of the jobs are handled locally. GSM-RUA performs significantly better than SMRA and ACRA when $t = 80$. Because it simultaneously optimizes resource allocation and device-side task splitting, GSM-RUA runs at its best and contributes significantly to the reduction of energy consumption in local computing.

The average queue backlogs for Q_n^L and Q_n^O with time slots shown in Figs. 3 and 4, respectively. Since SMRA only takes energy consumption optimization into account and is unable to address high-dimensional optimization that involves large state and action space challenges, it performs poorly when it comes to task offloading, offloading fewer tasks from Q_n^O to the server than GSM-RUA and ACRA. As consequently, SMRA has a larger average backlog of Q_n^O. Offloading more subtasks from devices to edge servers can reduce queue backlog owing to the queue-agnostic and stable computing resource allocation methodology.

The average queuing delay of Q_n^L and Q_n^O with time slots, respectively, are displayed in Fig. 5 and 6. According to the simulation results, Q_n^L and Q_n^O queuing delays are significantly reduced by GSM-RUA when compared to ACRA. Due to the acquisition of queue awareness and combined optimization of resource allocation and task offloading, GSM-RUA performs better when it comes to queuing delays.

In Fig. 7, for $Y = 10$ users and $P = 10$ B2B pairings, the average effective energy consumption performance is shown against the maximum B2B transmission distance d_{max}. The suggested approach achieves the best effective performance across the entire region, according to simulation findings. Concerning the random power allocation method and the power greedy algorithm, the suggested approach outperforms both by 135% and 208%, respectively, when $d_{max} = 30$m. The reason why random assignment performs second best is because it has a larger likelihood of consuming more energy than power-hungry algorithms, which always make the best use of any available power. The gain of the SE algorithm brought about by raising transmit power is insufficient to offset the accompanying effective energy loss. There are two reasons why the power-greedy algorithm performs the least efficiently in terms of energy use among the three. First, when allocating resources, electricity use is completely disregarded. Furthermore, in an environment with little interference, boosting transmit power beyond the node where the SE algorithm performs best results in

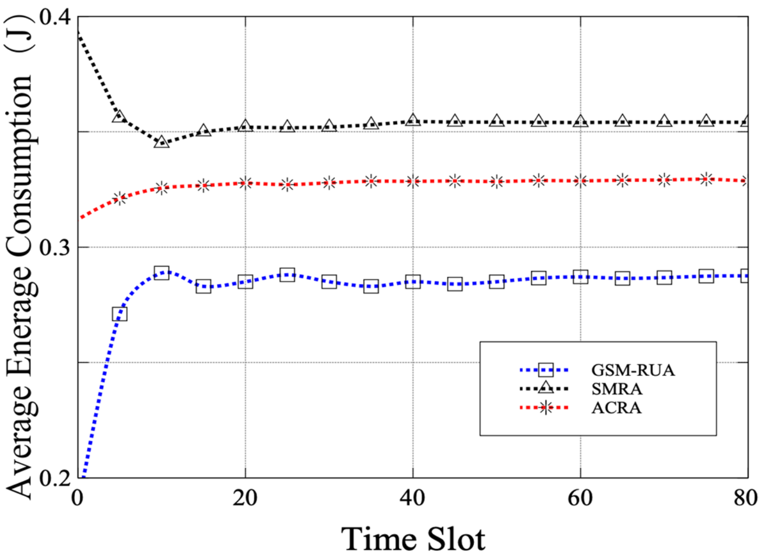

Fig. 2 Average energy consumption

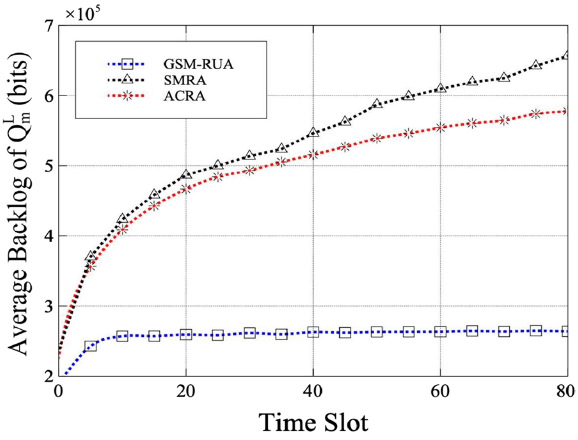

Fig. 3 Average backlog of Q_m^L

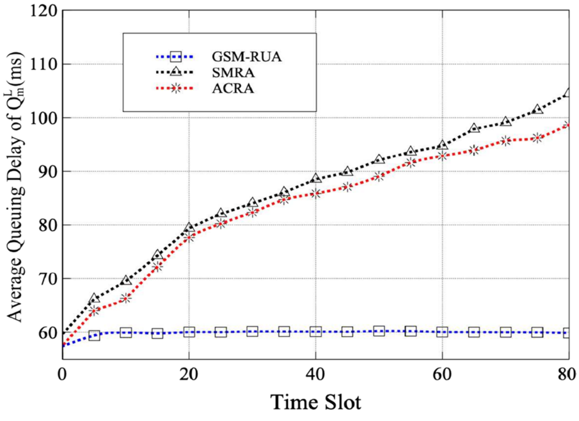

Fig. 5 Average queuing delays of Q_m^O

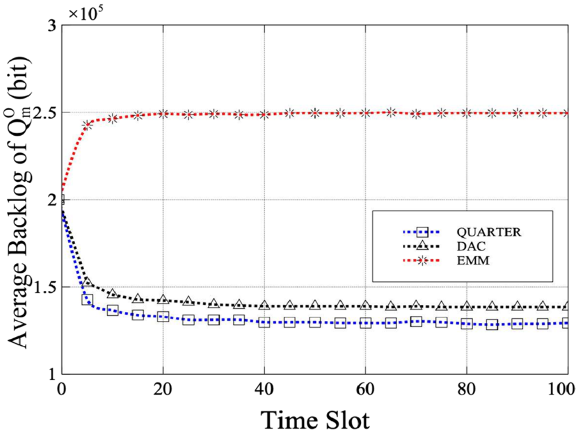

Fig. 4 Average backlog of Q_m^O

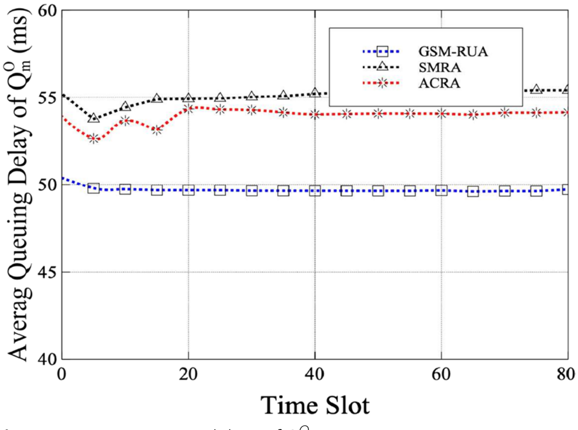

Fig. 6 Average queuing delays of Q_m^O

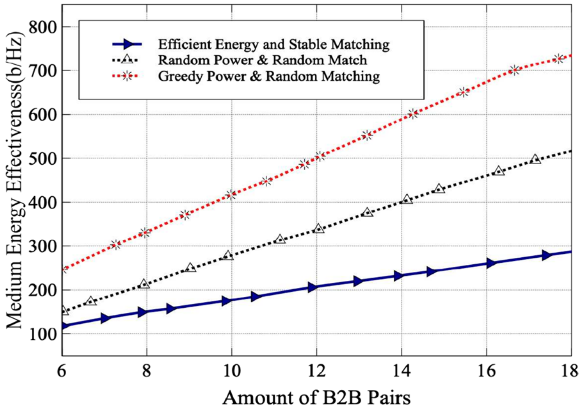

Fig. 7 Maximum B2B transmission distance and average effective energy consumption of a B2B pair

neither an enhancement of the SE algorithm nor a notable reduction in effective energy usage. It is important to note that when B2B transmission distance increases, all algorithms' effective energy performance declines since more transmission power is needed to keep the QoS performance at the same level as in the short-distance situation.

When $d_{max} = 30$m, Fig. 8 illustrates the relationship between the average effective energy consumption performance of B2B pairs and the number Y of active users and the number P of B2B pairs. The number of user and B2B pair activations grows linearly in relation to the average effective energy performance of all algorithms. The explanation is that as the number of users grows, there are more orthogonal channels overall and each B2B pair has more options than the original B2B pair in the expanded matching market. There is a higher chance that B2B pairs will be matched with superior relationships in the wider matching market. In comparison to the heuristic algorithm, the proposed method has the highest slope, indicating that it may gain more from a variety of options. The power-greedy algorithm has the smoothest slope since the benefits of selection variety are not completely utilized and power consumption is not taken into account when allocating resources.

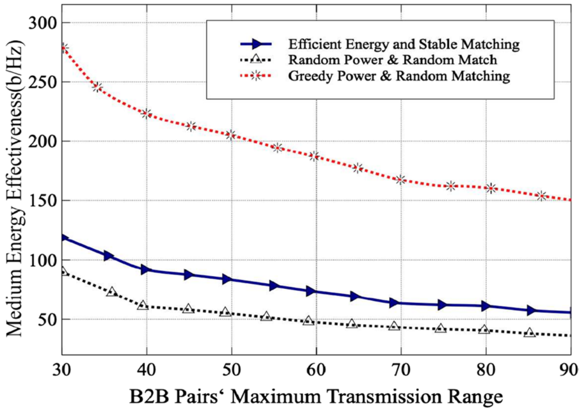

Fig. 8 Average effective energy consumption of B2B pairs vs. number of active B2B pairs and users

Discussion

The impact of the research results on the current state-of-the-art resource management algorithms and their applicability in real-life e-commerce scenarios can be analyzed from the following aspects:

(1) First of all, the impact of the research results on the current most advanced resource management algorithms is mainly reflected in algorithm design and optimization. The innovation of the GSM-RUA algorithm in resource unit allocation and group switching matching may inspire other researchers to improve existing resource management algorithms or propose new algorithms.
(2) Secondly, e-commerce scenarios usually face challenges such as high concurrency, low latency, and dynamic changes in resources. The GSM-RUA algorithm helps e-commerce platforms improve system throughput and reduce latency by optimizing resource allocation and handover matching, thereby improving user experience and platform performance. Then it may be widely used in e-commerce scenarios.

Conclusion

NOMA technology further increases spectral efficiency by enabling multiple users to perform non-orthogonal transmission on the same time and frequency resources. Edge computing and the combination of NOMA technology can intelligently select offloading strategies based on task characteristics and requirements. Edge computing already gives e-commerce platforms powerful computing capabilities and low-latency services.

This work investigates the joint optimization of resource units, large-connection computing resources, and task splitting on e-commerce platforms using NOMA. Based on energy consumption and queue backlogs, our suggested technique is intended to dynamically optimize multi-dimensional resource allocation to reduce energy consumption. To maximize the achievable effective energy usage under the highest transmit power, the cooperation problem is constructed while taking the user's preferences into account. The ideal energy usage for a certain match is modeled as its user's choice. The experimental results demonstrate that our proposed algorithm effectively achieves the trade-off between complexity and network capability, outperforming the current SMRA and ACRA in the Q_m^L queue backlog and in the Q_m^O queue backlog, respectively, A fresh thought for the research topic of allocating resources for B2B communication is also offered by the matching approach. Future research will investigate how to organize devices in e-commerce more effectively. Additionally, it incorporates the expansion of one-to-many matching, user preference modeling from a big data standpoint, and content caching with context awareness.

Authors' contributions

Xiao Zheng and Muhammad Tahir: Writing-original draft, conceptualization, methodology, data curation, visualization, writing-review, and editing. Khursheed Aurangzeb: supervision, resources, project administration, validation, writing, review, and editing. Muhammad Shahid Anwar: Resources, Formal Analysis, Writing-Review and Editing, Resources. Muhammad Aamir: supervision, project administration, writing, review, and editing. Ahmad Farzan: Resources, Validation, Writing, Review, and Editing. Rizwan Ullah: formal analysis, supervision, writing-review and editing, Project administration.

Funding

This Research is funded by Researchers Supporting Project Number (RSPD2024R947), King Saud University, Riyadh, Saudi Arabia.

Availability of data and materials

No datasets were generated or analysed during the current study.

Declarations

Competing interests

The authors declare no competing interests.

Author details

[1]School of Computer Science and Technology, Shandong University of Technology, Xincun West Road, Zibo, Shandong 255090, P.R. China. [2]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. [3]Department of Computer Science, Mohammad Ali Jinnah University, Block 6, P.E.C.H.S, Karachi 75400, Pakistan. [4]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, P. O. Box 51178, Riyadh 11543, Saudi Arabia. [5]Department of AI and Software, Gachon University, Seongnam-si 13120, South Korea. [6]College of Computer Science, Huanggang Normal University, Huanggang 438000, China. [7]Youwe Digital Agency, Kerry Hill, Horsforth, Leeds LS18 4JR, UK. [8]Department of Electrical Engineering, Chulalongkorn University, Bangkok 10330, Thailand.

Received: 14 March 2024 Accepted: 13 June 2024
Published online: 24 June 2024

References

1. Panek G, Fajjari I, Tarasiuk H et al (2024) Application Relocation in an Edge-Enabled 5G System: Use Cases, Architecture, and Challenges[J]. IEEE Commun Mag 60. https://doi.org/10.1109/MCOM.001.2100623
2. Abbas N, Zhang Y, Taherkordi A, Skeie T (2018) Mobile edge computing: A survey. IEEE Internet Things J 5(1):450–465
3. Alamri FS, Haseeb K, Saba T, Lloret J, Jimenez JM (2023) Multimedia IoT-surveillance optimization model using mobile-edge authentic computing. In: Arizona State University. https://doi.org/10.3934/mbe.2023847
4. Qin Z, Wang H, Qu Y, et al (2021) Air-Ground Collaborative Mobile Edge Computing: Architecture, Challenges, and Opportunities[J]. https://doi.org/10.48550/arXiv.2101.07930
5. Abbas S, Boulila W, Driss M, Victor N, Sampedro GA, Abisado M, Gadekallu TR (2023) Aspect Category Detection of Mobile Edge Customer Reviews: A Distributed and Trustworthy Restaurant Recommendation System. In: Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/TCE.2023.3323334
6. Rehman A, Saba T, Haseeb K, Alam T, Llore J (2022) Sustainability Model for the Internet of Health Things (IoHT) Using Reinforcement Learning with Mobile Edge Secured Services. In: MDPI. https://doi.org/10.3390/su141912185
7. Yu X, Li Q, Xie M, et al (2020) Performance of Uplink Multicell Multiuser Massive SM-MIMO Systems With Imperfect CSI and Pilot Contamination[J]. IEEE Syst J PP(99):1–12
8. Hu Y, Liu R, Kaushik A, Thompson J (2021) Performance Analysis of NOMA Multicast Systems Based on Rateless Codes with Delay Constraints. IEEE Trans Wirel Commun 20(8):5003–5017
9. Kiani A, Ansari N (2018) Edge computing aware noma for 5g networks. IEEE Internet Things J 5(2):1299–1306
10. Ding Z, Xu J, Dobre OA, Poor HV (2019) Joint power and time allocation for noma-mec offloading. IEEE Trans Veh Technol 68(6):6207–6211
11. Liu CF, Bennis M, Debbah M, Poor HV (2019) Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing. IEEE Trans Commun 67(6):4132–4150
12. Mao Y, You C, Zhang J, Huang K, Letaief KB (2017) A survey on mobile edge computing: The communication perspective. IEEE Commun Surv Tutorials 19(4):2322–2358
13. Akhlaqi MY, Hanapi ZBM (2023) Task offloading paradigm in mobile edge computing-current issues, adopted approaches, and future directions[J]. J Netw Comput Appl 212:103568
14. Luo F (2020) Advanced Cyber-Physical Infrastructures of Next-Generation Grids with Big Data Penetrations. J Energy Eng 146(1):02019002
15. Mcbride GD, Sumbwanyambe M (2020) Design and Construction of a Hybrid Edge-Cloud Smart Surveillance System with Object Detection[C]//International Conference on Computing, Communication and Intelligent Systems. IEEE. https://doi.org/10.1109/ICCCIS51004.2021.9397212
16. Sun X, Wang S, Xia Y et al (2020) Predictive-Trend-Aware Composition of Web Services With Time-Varying Qualityof-Service[J]. IEEE Access 8:1910–1921. https://doi.org/10.1109/ACCESS.2019.2962703
17. Premalatha B, Prakasam P (2024) A Review on FoG Computing in 5G Wireless Technologies: Research Challenges, Issues and Solutions[J]. Wireless Personal Communications 134(4):2455–2484. https://doi.org/10.1007/s11277-024-11061-y
18. Sarker VK, Queralta JP, Gia TN et al (2019) A Survey on LoRa for IoT: Integrating Edge Computing[C]//International Conference on Fog and Mobile Edge Computing. IEEE https://doi.org/10.1109/FMEC.2019.8795313
19. Basu D, Hussain AA, Hasan SF (2017) A distributed mechanism for Software-based mobility management[C]//2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS). IEEE https://doi.org/10.1109/ICSESS.2016.7883076
20. Enzo B, Michele S, Alireza M (2018) Fog-supported delay-constrained energy-saving live migration of VMs over MultiPath TCP/IP 5G connections[J]. IEEE Access 6:42327–42354. https://doi.org/10.1109/ACCESS.2018.2860249
21. Hong Z, Chen W, Huang H et al (2019) Multi-hop Cooperative Computation Offloading for Industrial IoT-Edge-Cloud Computing Environments[J]. IEEE Trans Parallel Distrib Syst (99):11–24. https://doi.org/10.1109/TPDS.2019.2926979
22. Zhou Z, Yu H, Mumtaz S, Al-Rubaye S, Tsourdos A, Hu RQ (2020) Power control optimization for large-scale multi-antenna systems. IEEE Trans Wirel Commun 19(11):7339–7352
23. Alameddine HA, Sharafeddine S, Sebbah S, Ayoubi S, Assi C (2019) Dynamic task offloading and scheduling for lowlatency iot services in multi-access edge computing. IEEE J Sel Areas Commun 37(3):668–682
24. Nouri N, Entezari A, Abouei J et al (2020) Dynamic Power–Latency Tradeoff for Mobile Edge Computation Offloading in NOMA-Based Networks. IEEE Internet Things J 7(4):2763–2776. https://doi.org/10.1109/JIOT.2019.2957313
25. Liao H, Zhou Z, Zhao X, Wang Y (2021) Learning-based queue-aware task offloading and resource allocation for space-air-ground-integrated power iot. IEEE Internet Things J 8(7):5250–5263
26. Zhou Z, Guo Y, He Y, Zhao X, Bazzi WM (2019) Access control and resource allocation for m2m communications in industrial automation. IEEE Trans Ind Inform 15(5):3093–3103

27. Liao H, Zhou Z, Zhao X, Zhang L, Mumtaz S, Jolfaei A, Ahmed SH, Bashir AK (2020) Learning-based context-aware resource allocation for edge-computing-empowered industrial iot. IEEE Internet Things J 7(5):4260–4277
28. Zeng M, Yadav A, Dobre OA, Poor HV (2019) Energyefficient joint user-rb association and power allocation for uplink hybrid noma-oma. IEEE Internet Things J 6(3):5119–5131
29. Harris DG (2019) Distributed Local Approximation Algorithms for Maximum Matching in Graphs and Hypergraphs[C]//Foundations of Computer Science. IEEE Comput https://doi.org/10.1109/FOCS.2019.00048
30. Baidas MW, Alsusa E, Al-Farra M et al (2020) Correction to: Multi-relay selection in energy-harvesting cooperative wireless networks: game-theoretic modeling and analysis[J]. Springer Science and Business Media LLC (2). https://doi.org/10.1007/S11235-019-00627-Y

Publisher's Note