Wireless Networks (2024) 30:557–576
https://doi.org/10.1007/s11276-023-03498-z

ORIGINAL PAPER

Joint optimization algorithm of offloading decision and resource allocation based on integrated sensing, communication, and computation

Shuo Sun[1,2] · Qi Zhu[1,2]

Accepted: 4 September 2023 / Published online: 25 September 2023

Abstract
Sixth-generation wireless systems not only have more demanding communication requirements, they are also expected to have high-precision sensing capabilities and sufficient computing power. Integrated sensing, communication, and computation (ISCC) can meet the above system requirements and save spectrum resources. In this paper, we build a resource allocation and offloading decision problem in an ISCC scenario that makes considerations for user mobility and partial offloading policies. The established problem minimizes the average task cost when given constraints such as the typical sensing failure rate and task completion delay. We use Lyapunov optimization theory to transform the proposed problem and propose a two-level optimization algorithm based on matching theory to offer a solution for the transformed problem. The inner layer obtains the task offloading ratio through theoretical derivation, and the outer layer determines the base station access and channel assignment based on the inner layer results. The simulation results show that the average task cost can be effectively reduced while also guaranteeing high-quality sensing performance.

Keywords ISCC · MEC · Lyapunov optimization · Matching theory · Service migration

1 Introduction

Mobile edge computing (MEC) is a technology that provides computing, storage, and communication capabilities at the network edge [1]. By bringing processing power closer to the end user, MEC can greatly reduce transmission delays, enhance computing capabilities, and support delay-sensitive and computing-intensive applications, such as autonomous driving, virtual reality, and video stream analysis. As a result, in MEC networks, the primary issues addressed in making offloading decisions are what to offload, how to offload it, and how much to offload. Resource allocation focuses on jointly managing the communication, computing, and storage resources of the system. Due to user offloading decisions and resource allocation having a significant and direct effect on system performance, studies in this area have recently become a research hotspot.

Due to the development of millimeter wave communication, the spectrums used in both communication and radar sensing have gradually begun to overlap. As a key technology in future mobile communication [2], integrated communication and sensing (ISAC) integrates the hardware of sensing and communication, allowing them to work in the same frequency range, and uses this integrated signal to offer simultaneous sensing and communication, thus making more efficient use of strained spectrum resources.

Future wireless networks are not only expected to have more demanding communication requirements, but will also need to have sufficient computing power and highly accurate sensing capabilities [3]. As a new framework, integrated sensing, communication, and computation

✉ Qi Zhu
zhuqi@njupt.edu.cn

Shuo Sun
17797712973@163.com

[1] Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210000, China

[2] Engineering Research Center of Health Service System Based on Ubiquitous Wireless Networks, Nanjing University of Posts and Telecommunications, Ministry of Education, Nanjing 210000, China

(ISCC) combines the advantages of both MEC and ISAC. ISAC is able to provide wireless network systems with sufficient computing power while also ensuring high accuracy and high resolution sensing capabilities that can be used in a number of intelligent automation applications such as autonomous driving [4]. ISCC can help self-driving vehicles to upload and process their generated tasks more effectively, while also accurately sensing other vehicles in their vicinity.

In reference [4], the authors established for the first time an implementation framework for ISCC. Through careful consideration of user throughput, task completion delay, and mutual information (MI) it is able to effectively strike a balance between communication, sensing, and computation. In this paper, we add considerations for user mobility, energy consumption, and partial offloading strategies based on the model provided by reference [4]. Taking user mobility into account will allow this paper to better analyze how the benefits of ISCC can be made applicable to mobile devices. Energy consumption is also a very important indicator for users in practical applications, and optimization of energy consumption can improve their service experiences. In addition, a partial offloading strategy can make the offloading of tasks more flexible for users. Finally, after taking these factors into account, an efficient resource allocation and offloading decision algorithm is developed.

The main contributions of this paper are summarized as follows:

(1) In this paper, we build a user cost minimization problem in a multi-BS multi-user ISCC scenario, making considerations for user mobility and partial offloading policies. The established problem involves user delay, energy consumption, migration cost, and sensing.

(2) The Lyapunov optimization theory is applied to transform the proposed problem into a single time slot deterministic optimization problem and a two-layer algorithm is used to solve the converted problem. The inner layer of the algorithm obtains the optimal offloading ratio by theoretical derivation via BS access and channel selection. Based on matching theory, the outer layer of the algorithm determines the user's BS access and channel selection according to the results of the inner layer.

(3) Simulation results show that the proposed algorithm in this paper achieves lower average task cost, completion delay, and energy consumption all while balancing task cost and sensing failure rate compared to the algorithm in reference [4] and other baseline algorithms.

The remainder of the article is organized as follows: a review of related literature is described in Sect. 2. The system model is explained in Sect. 3. The proposed algorithm is explained in detail in Sect. 4. The simulation results are shown and analyzed in Sect. 5. Finally, a conclusion based on the results of the simulation is offered in Sect. 6.

2 Related work

In the MEC scenario, reasonable allocation of user offloading decision and resource can effectively utilize the resources of users and edge nodes, thus greatly reducing the cost of system. In reference [5], the authors designed an effective user offloading decision and resource allocation algorithm in the scenario of single user and single edge server to minimize user energy consumption and task completion delay, respectively. In reference [6, 7], the authors directed at user delay and energy consumption jointly optimize offloading decisions, channel allocation, computation resource allocation and transmission power, and solved the established nonlinear optimization problem in the scenario of multi-user and single edge server. In reference [8], the authors used convex optimization and deep Q network to tackle the problem of offloading decision and resource allocation in multi-user and multi-server MEC networks with time-varying fading channels. In reference [9], the authors studied the task assignment problem of the optimal server in the scenario of multiple BS and multiple edge servers, proposed an offloading decision method in view of guaranteed quality of service, and minimized the overall energy consumption of edge computing networks. Device to device (D2D) communication has also been widely used in MEC. Users who generate tasks transmit tasks to idle user equipment through D2D to enhance their ability to process tasks. In MEC scenarios supporting D2D, the authors explored resource scheduling for multi-user cooperation and partial offloading to address the issue of latency and energy utilization in [10, 11].

User mobility will affect the wireless environment between the user and the edge node accessed. The quality of service for the user will drastically decrease as the distance between them and the base station (BS) grows. Under the single BS multi-user MEC scenario, the authors in [12] assumed that the user mobility is known. They proposed a heuristic offloading algorithm to address the issue of offloading decision and resource allocation. In reference [13], the authors defined the ideal offloading node selection strategy as a Markov decision process and solved it by value iterative method based on the location of mobile users and the bandwidth of heterogeneous edge servers. In reference [14], the authors proposed a dual time scale mobility management framework considering the difference in time scales between the dynamic changes of user task arrival and user location changes and the service migration caused by user mobility. In reference[15], the authors estimated the mobility of vehicles when 5G users move across wireless cells and used this prediction and online algorithms to determine when and where to migrate computing services (containers or virtual machines (VMs)) running on edge nodes to ensure the continuity of vehicle services. In [16], under the MEC scenario supporting D2D, by considering user mobility, distribution of resources, task properties and energy limits of user equipment, a task power allocation method was developed to reduce task completion delay. In reference [17], the authors solved the problem of the average total cost minimization over a long period of time by optimizing the task division, transmission power, CPU processing frequency, the harvested energy, and the association vector of Internet of Things devices.

At present, ISAC research work is devoted to s the spectrum allocation, the design of transceiver structure and frame structure, ISAC integrated waveform design, joint coding design, temporal-spectral-spatial signal processing [2]. In reference [18], the authors studied the spectrum allocation problem between SSPs and terminal devices configured with integrated radar and communication systems using a game-theoretic approach. In reference [19], based on information theory, the authors put forward an adaptive orthogonal frequency division multiplexing (OFDM) waveform design mode which considered integrated radar and communication waveform. In reference [20], the authors selected subcarriers for communication and sensing under the target mutual information as well as communication rate constraint and assigned frequencies on the subcarriers, which ultimately minimized the total radiated power of the system.

The ISCC research has just started and there is not much related work yet. In [4], the authors combined MEC paradigm and ISAC technology to build an ISCC implementation framework. This framework first attempted to combine the advantages of ISAC and MEC to support automated applications. In [21], the authors derived two typical beamforming methods by considering communication, sensing, and computation. The goal was to minimize the total transmission power and maximizing the weighted overall performance, respectively.

3 System model

The model is presented in Fig. 1. There are N BSs, M users, and L channels in the system. The BSs set is represented as $\mathcal{N} = \{1, 2, 3, \ldots, N\}$, the users set is represented as $\mathcal{M} = \{1, 2, 3, \ldots, M\}$, and the channels set is represented as $\mathcal{L} = \{1, 2, 3, \ldots, L\}$. All BSs share multiple channels. Each BS is equipped with a server to handle the tasks offloaded by users. In the current mobile edge computing scenario, the storage resources of the base station are still relatively abundant. Therefore, in this paper, we do not consider the storage resource limit of the base station [22, 23]. When generating a computing task to be processed, the user offloads partially task to the BS server for processing, and the integrated signal is used to sense the environment around the user when the user sends the task. The blue solid line in Fig. 1 represents the integrated signal used by users for communication and sensing, the blue dotted line represents the echo of the integrated signal reflected by environmental objects, and the red dotted line represents the interference of other users' integrated signals to the current user.

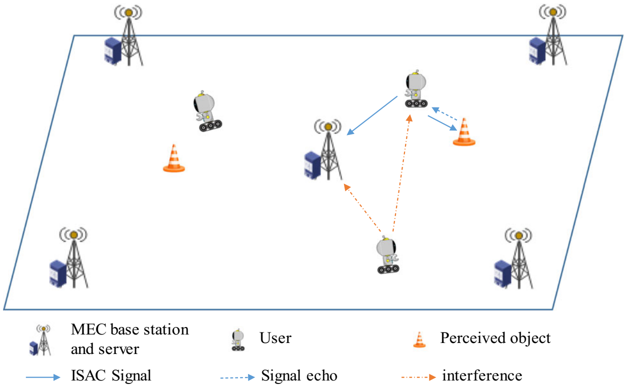


Fig. 1 System model

The mobility of users is considered in this paper. The user's mobility process is divided into discrete time slots. The length of a single time slot is denoted using τ. The time slots are denoted by the set $T = \{1, 2, 3, \ldots, t, \ldots\}$. It is assumed that the user's position within a time slot remains unchanged, and the position in different time slots is different. Each user has a service, which is composed of a series of tasks, and a dedicated VM or container is built on the edge server to process the generated tasks. With the user's movement, the VM or container will migrate with the handover of the user's connected BS [24].

3.1 Task model

In this paper, it is considered that each user generates one task in a single time slot. At the beginning of each time slot, each user generates a task whose data volume follows uniform distribution. The task is represented by tuple $\langle I_m(t), r_m(t), D_m(t) \rangle$, where $I_m(t)$ represents the size of the data volume to be communicated by the task generated by user m in time slot t, and the unit is *bit*. $C_m(t)$ represents the task calculation load of user m in time slot t, namely, the number of cycles the task needs the computer CPU to run to calculate, and the unit is *cycle*, specifically expressed as $C_m(t) = I_m(t)r_m$, where r_m represents the user m's task calculation intensity, namely, the number of cycles the computer CPU needs to run for each bit of data to calculate. $D_m(t)$ is the maximum delay requirement of the task generated by user m in time slot t. This paper considers that the maximum delay requirement of the task is less than or equal to the time slot length, that is, $D_m(t) \le \tau$ [17]. This paper considers the partial offloading mechanism. The task offloading ratio of user m in time slot t is expressed as $\rho_m(t)$, and the tasks are assumed to be randomly separable. However, this paper considers that users need to sense the surrounding environment in each time slot, so it is considered that the task offloading ratio has a minimum requirement $\rho_{\min}$.

3.2 Communication model

The user selects a BS for access in each time slot, and sends the tasks to the BS through the channel allocated by the BS. The symbol $a_m^n(t)$ indicates the association between user m and BS n in time slot t, namely, when user m accesses BS n in time slot t, $a_m^n(t) = 1$, and vice versa. All BSs share L channel, and each channel has the same bandwidth, which is W. The signals of users using the same channel will interfere with each other. The symbol $b_m^l(t)$ indicates whether user m occupies channel l in time slot t, namely, when user m occupies channel l in time slot t, $b_m^l(t) = 1$, and vice versa. Users can only access one BS in each time slot, occupying one channel, that is

$$\sum_{n\in\mathcal{N}} a_m^n(t) = 1 \quad \forall m \in \mathcal{M}, t \in T \tag{1}$$

$$\sum_{l\in\mathcal{L}} b_m^l(t) = 1 \quad \forall m \in \mathcal{M}, t \in T \tag{2}$$

One channel of a BS can be used by at most one user, that is

$$\sum_{m\in\mathcal{M}} a_m^n(t) b_m^l(t) = 1 \, \forall n \in \mathcal{N}, l \in \mathcal{L}, t \in T \tag{3}$$

The large-scale fading and small-scale fading of the channel between the user and the BS are considered at the same time. The small-scale fading of the channel follows the Rayleigh distribution. Channel gain is constant within a time slot but *i.i.d.* in different time slots. The symbol $h_{m,n}^l(t)$ is used to represent the channel gain between the user m and the BS n in time slot t and on channel l. When user m offloads task to the BS during time slot t, the transmission power is P_m, and the data transmission rate of user m is expressed as

$$R_m^{Tr}(t) = \sum_{n\in\mathcal{N}} \sum_{l\in\mathcal{L}} a_m^n(t) b_m^l(t) W \log_2\left(1 + \Gamma_{m,n}^l(t)\right) \tag{4}$$

where $\sigma^2 = N_0 W$ is the channel noise power, and $\Gamma_{m,n}^l(t) = \frac{P_m h_{m,n}^l(t)}{\sum_{m'\in\mathcal{M}\setminus\{m\}} b_{m'}^l(t) P_{m'} h_{m',n}^l(t) + \sigma^2}$ represents SINR of communication when user m occupies channel l of BS n.

The time consumed by user m to transmit the task in time slot t is calculated as

$$T_m^{Tr}(t) = \frac{I_m(t)\rho_m(t)}{R_m^{Tr}(t)} \tag{5}$$

The energy consumption for the user m to transmit the task in time slot t is

$$E_m^{Tr}(t) = P_m T_m^{Tr}(t) \tag{6}$$

Data volume in the task calculation results is small, and the return time can be ignored.

3.3 Computational model

The tasks generated by users in each time slot will be divided into two parts. One part will be transmitted to the BS through wireless channel for processing, and the other part will be processed locally. The symbol $f_{m,local}$ is used to indicate the computing power of user m, and the unit is *Hz*. Therefore, the local calculation time of user m's task in time slot t is given by

$$T_m^{local}(t) = \frac{C_m(t)(1 - \rho_m(t))}{f_{m,local}} \tag{7}$$

The energy consumption for user m to process tasks locally in time slot t is expressed as [25]

$$E_m^{local}(t) = \mathcal{K}(1 - \rho_m(t)) C_m(t) f_{m,local}^2 \tag{8}$$

where $\mathcal{K}$ refers to the power coefficient, which is decided by the chip structure.

After receiving the task transmitted by the user, the BS divides all computing resources equally among all users' computing tasks. Then the computing resources allotted to user m by BS n in time slot t are denoted by

$$f_m^n(t) = \frac{a_m^n(t) F_n}{\sum_{m\in\mathcal{M}} a_m^n(t)} \tag{9}$$

where F_n denotes the computing resources of BS n.

The computing resources obtained by user m in time slot t are denoted by

$$f_m(t) = \sum_{n\in\mathcal{N}} f_m^n(t) \tag{10}$$

The task calculation delay of user m in time slot t is

$$T_m^{com}(t) = \frac{C_m(t)\rho_m(t)}{f_m(t)} \tag{11}$$

3.4 Sensing model

When sending the offloading task to the BS, the user terminal equipment with ISAC analyzes the integrated signal echo reflected by the surrounding objects to sense the surrounding environment. This paper uses the conditional MI of echo signal and channel impulse response (the channel is the channel between the user and the random sensing target) to measure the performance of the user's sensing of the surrounding environment [19].

In ISAC scenarios, OFDM-based integrated waveforms for radar communication are often employed. MI indicates the reduction of information uncertainty after observation. In OFDM-based radar target identification and classification, MI is a commonly used accuracy metric of sensing.

$s_m^l(\xi)$ is used to denote the waveform used by user m on channel l at time slot t. $\phi_m^l(\xi)$ is used to denote the impulse response between user m and the sensed target on channel l at time slot t. $z_m^l(\xi)$ is used to denote the reflected signal received by user m on channel l at time slot t. The MI of user m in time slot t is given by

$$\begin{aligned} MI_m(t) &= \sum_{l\in\mathcal{L}} I\left(z_m^l(\xi);\phi_m^l(\xi)|s_m^l(\xi)\right) \\ &= \sum_{l\in\mathcal{L}} b_m^l(t)\frac{1}{2}KT_sW\log_2\left(1+\Gamma_{m,rad}^l(t)\right) \end{aligned} \tag{12}$$

where $\Gamma_{m,rad}^l(t)$ represents the radar the radar SINR of user m in time slot t and on channel l, specifically

$$\Gamma_m^l(t) = \frac{P_mKT_s^2|V_m^l(f^l)|^2}{\sum_{m'\in\mathcal{M}\setminus\{m\}} b_{m'}^l(t)P_{m'}h_{m',m}^l(t)+\sigma^2} \tag{13}$$

where, K is the number of symbols of the integrated signal, T_s represents the duration of a symbol, and $V_m^l(f^l)$ represents the Fourier transform of the channel impulse response at the frequency f^l of channel l, and its value is related to the antenna gain, radar cross section, and channel gain during transmission and reception. Here, it is assumed that it obeys the standard normal distribution [19]. $h_{m',m}^l(t)$ represents the channel gain on channel l between user m' and user m.

A minimum requirement of MI ($MI_{\min}$) is needed to ensure the accuracy of the user's estimate of the target impulse response. When the user's MI is less than the minimum requirement, the user is determined to have failed sensing. The user's sensing failure event is represented as

$$S_m(t) = indicator_{\{MI_m(t)<MI_{\min}\}} \tag{14}$$

where, $indicator_{\{x\}}$ represents the indicator function, namely, when x is true, the value of the indicator function is 1, and vice versa.

3.5 Service migration model

With the movement of users, some users will be farther and farther away from the connected BS. At this time, the problem of user handover will occur. If the user switches to a new BS, the service of the user on the BS side needs to migrate with the handover of the user, and the migration of the user service result in migration cost. In this paper, the migration cost is considered in two main parts, the latency and the other resource consumption caused by the migration of VMs. The other resource consumption caused by the migration of user m in time slot t is expressed as [17]

$$c_m^{mi}(t) = \sum_{n\in\mathcal{N}}\frac{\varepsilon}{2}\left[(1-a_m^n(t-1))a_m^n(t)+(1-a_m^n(t))a_m^n(t-1)\right] \tag{15}$$

where ε represents the resource consumption of migration, which is assumed to be a fixed value in this paper.

The migration delay of user m in slot t is denoted by

$$T_m^{mi}(t) = \sum_{n\in\mathcal{N}}\frac{t_{mi}}{2}\left[(1-a_m^n(t-1))a_m^n(t)+(1-a_m^n(t))a_m^n(t-1)\right] \tag{16}$$

where t_{mi} denotes a fixed migration delay.

It can be seen from (15) and (16) that when the user selects a different BS in the current time slot from the one in the previous time slot, there will be a migration latency of size t_{mi} and resource consumption of size ε, and vice versa.

The task processing delay of user m in time slot t is the maximum value of local computing delay and BS processing delay, which is calculated as

$$T_m(t) = \max\left\{T_m^{local}(t), T_m^{mi}(t)+T_m^{Tr}(t)+T_m^{com}(t)\right\} \tag{17}$$

The energy consumption of user m in time slot t processing task is the sum of local processing energy consumption and data transmission energy consumption, which is calculated as

$$E_m(t) = E_m^{local}(t)+E_m^{Tr}(t) \tag{18}$$

3.6 Optimization problem

In the computing offloading scenario, users' task completion delay and energy consumption are often particularly important to users. Therefore, these two indicators are often used to measure the service quality of users. In this paper, the user cost function is defined by considering the delay, energy consumption and migration cost. The cost function of user m in slot t is given by

$$U_m(t) = \alpha T_m(t)+\beta E_m(t)+c_m^{mi}(t) \tag{19}$$

where α and β are the weight coefficients of delay and energy consumption, respectively.

In this paper, we consider minimizing the sum of users' long-term average costs. The optimization problem is expressed as:

$$
\begin{aligned}
&P1:\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{\rho}}\lim_{i\to\infty}\frac{\sum_{t=1}^{i}\sum_{m\in\mathcal{M}}E[U_m(t)]}{i}\\
&s.t.\,(C1)\sum_{n\in\mathcal{N}}a_m^n(t)=1,\quad \forall m\in\mathcal{M},\ t\in T\\
&(C2)\sum_{l\in\mathcal{L}}b_m^l(t)=1,\quad \forall m\in\mathcal{M},\ t\in T\\
&(C3)\sum_{m\in\mathcal{M}}a_m^n(t)b_m^l(t)=1,\quad \forall n\in\mathcal{N},\ l\in\mathcal{L},\ t\in T\\
&(C4)\lim_{i\to\infty}\frac{\sum_{t=1}^{i}E[S_m(t)]}{i}\le\delta,\quad \forall m\in\mathcal{M}\\
&(C5)T_m(t)\le D_m(t),\quad \forall m\in\mathcal{M},\ t\in T\\
&(C6)\rho_m(t)\in[\rho_{\min},1],\quad \forall m\in\mathcal{M},\ t\in T\\
&(C7)a_m^n(t),b_m^l(t)\in\{0,1\},\quad \forall m\in\mathcal{M},\ n\in\mathcal{N},\ l\in\mathcal{L},\ t\in T
\end{aligned}
\tag{20}
$$

In $P1$, the variable set $\boldsymbol{A}=\{a_m^n(t)|m\in\mathcal{M},\ n\in\mathcal{N},\ t\in T\}$ represents the access of each user to each BS in each time slot, $\boldsymbol{B}=\{b_m^l(t)|m\in\mathcal{M},\ l\in\mathcal{L},\ t\in T\}$ represents the occupation of each user to each channel in each time slot, and $\boldsymbol{\rho}=\{\rho_m(t)|m\in\mathcal{M},\ t\in T\}$ represents the task offloading ratio of each user in each time slot. Furthermore, constraints $(C1)$ and $(C2)$ indicate that a user can only select one BS for access and occupy one channel in each time slot. Constraint $(C3)$ means that a channel of a BS can be used by at most one user in one time slot. Constraint $(C4)$ shows that the average time of user sensing failure times should be less than a certain set value, that is, the user's long-term sensing failure rate should be below a certain threshold. The objective function in this thesis includes not only the average task latency, but also the average task energy consumption and migration cost. In order to prevent the optimisation results from being more biased towards energy consumption and migration costs, which would lead to greater latency, a constraint (C5) is added so that the user's task can be completed within the maximum latency requirement.Constraint $(C6)$ indicates that the tasks of each user in each slot can be randomly divided between the minimum value $\rho_{\min}$ and 1. Constraint $(C7)$ indicates that the BS selection and channel allocation variables are binary variables.

4 Joint optimization algorithm for offloading ratio, BS selection and channel allocation

In this section, the solution process of problem P1 will be explained. Based on the Lyapunov optimization theory, the problem P1 is transformed into a single time slot deterministic optimization problem and a two-layer algorithm is used to solve the transformed problem.

For the sensing failure event of user m in slot t, namely, $S_m(t)$, we define a corresponding virtual queue with an initial value of 0. We express the backlog of each queue as $S_m(t)$, where $Q_m(0)=0$. We use $S_m(t)$ to indicate the increase of virtual queue in slot t, and δ to indicate the decrease of virtual queue. The virtual queue backlog of user m in slot t and slot $t+1$ is satisfied

$$
Q_m(t+1)=[Q_m(t)+S_m(t)-\delta]^+ \tag{21}
$$

where $[x]^+$ means $\max\{x,0\}$. According to [26], by preserving the stability of the virtual queue, it can ensure the satisfaction of the long-term sensing failure rate constraint $(C4)$.

Vector $\boldsymbol{\theta}(t)=[Q_1(t),Q_2(t),\ldots,Q_M(t)]$ is used to represent the queue backlog of the whole system. To ensure the stability of the virtual queue, the conditional Lyapunov drift function of the queue backlog of the whole system is defined as

$$
\Delta(\boldsymbol{\theta}(t))=E\left[\frac{1}{2}\sum_{m\in\mathcal{M}}Q_m(t+1)^2-\frac{1}{2}\sum_{m\in\mathcal{M}}Q_m(t)^2|\boldsymbol{\theta}(t)\right] \tag{22}
$$

The Lyapunov drift function represents the difference between the changes of the quadratic function of the whole system queue backlog in the two time slots before and after. The Lyapunov drift plus penalty function is defined as

$$
\Delta_V(\boldsymbol{\theta}(t))=\Delta(\boldsymbol{\theta}(t))+VE\left[\sum_{m\in\mathcal{M}}U_m(t)|\boldsymbol{\theta}(t)\right] \tag{23}
$$

where V is the control parameter of queue stability and objective function, and its value is greater than 0.

The algorithm in this paper focuses on minimizing the long-term average cost as well as ensuring queue stability by optimizing an upper bound on the Lyapunov drift-plus-penalty function, i.e., satisfying the sensing failure rate constraint. Theorem 1 gives the upper bound of the Lyapunov drift-plus-penalty function.

Theorem 1 *For a feasible decision at any time, the Lyapunov drift plus penalty function satisfies the inequality.*

$$
\begin{aligned}
\Delta_V(\boldsymbol{\theta}(t))\le B+E\left[\sum_{m\in\mathcal{M}}Q_m(t)(S_m(t)-\delta)|\boldsymbol{\theta}(t)\right]\\
+VE\left[\sum_{m\in\mathcal{M}}U_m(t)|\boldsymbol{\theta}(t)\right]
\end{aligned}
\tag{24}
$$

where $B=\frac{1}{2}(M+M\delta^2)$.

Proof From (21), $Q_m(t+1)^2\le(Q_m(t)+S_m(t)-\delta)^2$, so.

$$
\begin{aligned}
\frac{1}{2}\sum_{m\in\mathcal{M}} Q_m(t+1)^2 &\le \frac{1}{2}\sum_{m\in\mathcal{M}}\left(Q_m(t)^2+S_m(t)^2+2Q_m(t)S_m(t)\right.\\
&\qquad \left.-2\delta\big(Q_m(t)+S_m(t)\big)+\delta^2\right)\\
&\overset{(a)}{\le} \frac{1}{2}\sum_{m\in\mathcal{M}}\left(Q_m(t)^2+S_m(t)^2\right.\\
&\qquad \left.+2Q_m(t)(S_m(t)-\delta)+\delta^2\right)
\end{aligned}
\tag{25}
$$

where (a) is because the values of δ, $S_m(t)$ are greater than or equal to 0.

From (25),

$$
\begin{aligned}
&\frac{1}{2}\sum_{m\in M} Q_m(t+1)^2-\frac{1}{2}\sum_{m\in\mathcal{M}} Q_m(t)^2\\
&\le \sum_{m\in\mathcal{M}}\left(\frac{1}{2}S_m(t)^2+Q_m(t)(S_m(t)-\delta)+\frac{\delta^2}{2}\right) \qquad (26)\\
&\overset{(b)}{\le} \frac{1}{2}(M+M\delta^2)+\sum_{m\in\mathcal{M}}\left(Q_m(t)(S_m(t)-\delta)\right)
\end{aligned}
$$

where (b) is because the value of $S_m(t)$ is 0 or 1.

From (26),

$$
\begin{aligned}
\Delta_V(t) &= \Delta(\boldsymbol{\theta}(t))+VE\left[\sum_{m\in\mathcal{M}} U_m(t)|\boldsymbol{\theta}(t)\right]\\
&\le \frac{1}{2}(M+M\delta^2)+\sum_{m\in\mathcal{M}} E(Q_m(t)(S_m(t)-\delta)|\boldsymbol{\theta}(t))\\
&\quad + VE\left[\sum_{m\in\mathcal{M}} U_m(t)|\boldsymbol{\theta}(t)\right]
\end{aligned}
\tag{27}
$$

Let $B=\frac{1}{2}(M+M\delta^2)$, Theorem 1 is proved.

By applying Theorem 1 and observing the channel state, user movement, task generation, queue backlog, and user BS selection for the current time slot, problem P1 is transformed into the following problem P2.

$$
\begin{aligned}
&P2: \min_{\boldsymbol{A}(t),\boldsymbol{B}(t),\boldsymbol{\rho}(t)} B+\sum_{m\in\mathcal{M}} Q_m(t)S_m(t)+V\sum_{m\in\mathcal{M}} U_m(t)\\
&s.t.\,(C1)\sum_{n\in\mathcal{N}} a_m^n(t)=1, \quad \forall m\in\mathcal{M}\\
&(C2)\sum_{l\in\mathcal{L}} b_m^l(t)=1, \quad \forall m\in\mathcal{M}\\
&(C3)\sum_{m\in\mathcal{M}} a_m^n(t)b_m^l(t)=1, \quad \forall n\in\mathcal{N},\, l\in\mathcal{L}\\
&(C5)T_m(t)\le D_m(t), \quad \forall m\in\mathcal{M}\\
&(C6)\rho_m(t)\in[\rho_{\min},1], \quad \forall m\in\mathcal{M}\\
&(C7)a_m^n(t),b_m^l(t)\in\{0,1\}, \quad \forall m\in\mathcal{M},\, n\in\mathcal{N},\, l\in\mathcal{L}
\end{aligned}
\tag{28}
$$

After the original Problem P1 is transformed into Problem P2, Theorem 2 explains the relationship between the optimal values before and after the problem is transformed.

Theorem 2 *Assume that the optimal solution, queue backlog, sensing failure event and user cost sum obtained by solving problem P2 are respectively expressed as* $\boldsymbol{a}^*(t)$, $Q_m^*(t)$, $S_m^*(t)$, $U_{sum}^*(t)=\sum_{m\in\mathcal{M}} U_m^*(t)$, $\forall m\in M$, $t=1, 2,\ldots$, U_{opt} *are the optimal values corresponding to the optimal value of original problem P1, then,*

$$
\lim_{i\to\infty}\frac{\sum_{t=1}^{i} E[U_{sum}^*(t)|\boldsymbol{\theta}(t)]}{i}\le U_{opt}+\frac{B}{V}
\tag{29}
$$

Proof According to Theorem 1:

$$
\begin{aligned}
&\frac{1}{2}\sum_{m\in\mathcal{M}} Q_m^*(t+1)^2-\frac{1}{2}\sum_{m\in\mathcal{M}} Q_m^*(t)^2+VU_{sum}^*(t)\\
&\le B+VU_{sum}^*(t)+\sum_{m\in\mathcal{M}}\left(Q_m^*(t)(S_m^*(t)-\delta)\right)\\
&\le B+V\sum_{m\in\mathcal{M}} U_m(\boldsymbol{w}(t),\boldsymbol{a}^{\cdot}(t))\\
&\quad +\sum_{m\in\mathcal{M}} Q_m^*(S_m(\boldsymbol{w}(t),\boldsymbol{a}^{\cdot}(t))-\delta)
\end{aligned}
\tag{30}
$$

Expectations for both ends of inequality (30):

$$
\begin{aligned}
&E\left[\frac{1}{2}\sum_{m\in\mathcal{M}} Q_m^*(t+1)^2-\frac{1}{2}\sum_{m\in\mathcal{M}} Q_m^*(t)^2+VU_{sum}^*(t)|\boldsymbol{\theta}(t)\right]\\
&\le E\left[B+V\sum_{m\in\mathcal{M}} U_m(\boldsymbol{w}(t),\boldsymbol{a}^{\cdot}(t))\right.\\
&\quad \left.+\sum_{m\in\mathcal{M}} Q_m^*(t)(S_m(\boldsymbol{w}(t),\boldsymbol{a}^{\cdot}(t))-\delta)|\boldsymbol{\theta}(t)\right]\\
&=B+VE\left[\sum_{m\in\mathcal{M}} U_m(\boldsymbol{w}(t),\boldsymbol{a}^{\cdot}(t))|\boldsymbol{\theta}(t)\right]\\
&\quad +E\left[\sum_{m\in\mathcal{M}} Q_m^*(t)(S_m(\boldsymbol{w}(t),\boldsymbol{a}^{\cdot}(t))-\delta)|\boldsymbol{\theta}(t)\right]\\
&=B+VE\left[\sum_{m\in\mathcal{M}} U_m(\boldsymbol{w}(t),\boldsymbol{a}^{\cdot}(t))\right]+\sum_{m\in\mathcal{M}} E\left[Q_m^*(t)|\boldsymbol{\theta}(t)\right]\\
&\quad * E[S_m(\boldsymbol{w}(t),\boldsymbol{a}^{\cdot}(t)-\delta|\boldsymbol{\theta}(t))]\\
&\le B+VU_{opt}
\end{aligned}
\tag{31}
$$

where $\boldsymbol{w}(t)$ denotes the random event (channel status, task size, user movement) in time slot t and $\boldsymbol{a}^{\cdot}(t)$ denotes the decision (BS selection, channel allocation, computing resource allocation) in the optimal $w-only$ policy.

The left and right sides of formula (31) are summed in i time slots to get:

$$\begin{aligned} i*(B+VU_{opt}) \geq & \sum_{t=1}^{i} E\left[\frac{1}{2}\sum_{m\in\mathcal{M}} Q_m^*(t+1)^2 - \right. \\ & \left. -\frac{1}{2}\sum_{m\in\mathcal{M}} Q_m^*(t)^2 + VU_{sum}^*(t)|\boldsymbol{\theta}(t)\right] \\ & = \sum_{t=1}^{i} E[L(\boldsymbol{\theta}(i)) - L(\boldsymbol{\theta}(0))|\boldsymbol{\theta}(t)] + VE[U_{sum}^*(t)|\boldsymbol{\theta}(t)] \\ & = \sum_{t=1}^{i} E[L(\boldsymbol{\theta}(i))|\boldsymbol{\theta}(t)] + VE[U_{sum}^*(t)|\boldsymbol{\theta}(t)] \\ & \geq V\sum_{t=1}^{i} E\left[U_{sum}^*(t)|\boldsymbol{\theta}(t)\right] \end{aligned} \tag{32}$$

So

$$\frac{1}{i}\sum_{t=1}^{i} E[U_{sum}^*(t)|\boldsymbol{\theta}(t)] \leq U_{opt} + \frac{B}{V} \tag{33}$$

Let $i \to \infty$, $\lim\limits_{i\to\infty} \frac{1}{i}\sum_{t=1}^{i} E[U_{sum}^*(t)|\boldsymbol{\theta}(t)] \leq U_{opt} + \frac{B}{V}$. Theorem 2 is proved.

Theorem 2 shows that when $V \to \infty$, the sum of long-term average user costs obtained by solving single time slot problem P2 can infinitely approximate the optimal value of the original problem P1.

Since the objective function of problem P2 is a complex nonlinear function, and the BS selection and channel allocation are binary variables, and the task offloading ratio is continuous variables, problem P2 is a mixed integer nonlinear programming problem. Such problems are commonly *NPhard* problems, and are unable to be solved by polynomial time complexity algorithm. With the aim of solving problem P2, this paper designs a joint optimization algorithm of task offloading ratio, BS selection and channel allocation. The algorithm divides the problem P2 into two layers, the internal layer obtains the optimal task offloading ratio through theoretical derivation, and the outer layer determines the BS selection and channel allocation through the matching method.

4.1 Task offloading ratio

When the BS selection and channel allocation of user are given, the internal layer problem is

$$\begin{aligned} P2.1: \min_{\boldsymbol{\rho}(t)} B + & \sum_{m\in\mathcal{M}} Q_m(t)S_m(t) + V\sum_{m\in\mathcal{M}} U_m(t) \\ & s.t.\ (C5) T_m(t) \leq D_m(t), \forall m \in \mathcal{M} \\ & (C6) \rho_m(t) \in [\rho_{\min}, 1], \forall m \in \mathcal{M} \end{aligned} \tag{34}$$

The objective function of question $P2.1$ is represented by the symbol f, which is detailed as follows:

$$\begin{aligned} f = B + & \sum_{m\in\mathcal{M}} (Q_m(t)S_m(t) \\ & + V\left(\alpha \max\left\{\underbrace{\frac{(1-\rho_m(t))C_m(t)}{f_{m,local}}}_{(1)}, \underbrace{\rho_m(t)(\frac{I_m(t)}{R_m^{Tr}(t)} + \frac{C_m(t)}{f_m(t)}) + T_{mi}}_{(2)}\right\}\right. \\ & \left.\left.+\beta\left(\underbrace{\mathcal{K}(1-\rho_m(t))C_m(t)f_{m,local}^2}_{(3)} + \underbrace{P_m\rho_m(t)\frac{I_m(t)}{R_m^{Tr}(t)}}_{(4)}\right)\right)\right) \end{aligned} \tag{35}$$

(1), (2), (3), (4) in (35) are linear functions of variable $\rho_m(t)$. Because the maximum function of the two convex functions is also a convex function, the objective function is a convex function of $\rho_m(t)$. Constraint $(C5)$ is obviously a convex function after moving the right term of the inequality to the left, so problem $P2.1$ is a convex problem.

Theorem 3 *Let* $F_0 = 1 - D_m(t)f_{m,local}/C_m(t)$ *and* $F_1 = (D_m(t) - T_m^{mi}(t))/\left(\frac{I_m(t)}{R_m^{Tr}(t)} + \frac{C_m(t)}{f_m(t)}\right)$. *When* $F_0 > F_1$ *or* $F_1 < \rho_{\min}$, *problem* $P2.1$ *has no solution, otherwise, the optimal solution of* $P2.1$ *is*:

$$\rho_m^*(t) = \begin{cases} \max\{F_0, \rho_{\min}\}, & \text{if } F_2 \geq 0 \text{ and } F_3 < \rho_{\min} \\ \max\{1, F_1\}, & \text{if } F_2 < 0 \text{ and } F_3 < \rho_{\min} \\ \rho_{m,1}(\text{t}), & \text{if } F_3 \geq \rho_{\min} \text{ and } f_1(m,t) \leq f_2(m,t) \\ \rho_{m,2}(t), & \text{if } F_3 \geq \rho_{\min} \text{ and } f_1(m,t) \geq f_2(m,t) \end{cases} \tag{36}$$

where

$$\rho_{m,1}(t) = \begin{cases} \max\{\rho_{\min}, F_0\}, & \text{if } F_4 \geq 0 \\ \min\{F_1, F_3, 1\}, & \text{if } F_4 < 0 \end{cases} \tag{37}$$

$$\rho_{m,2}(t) = \begin{cases} \max\{F_0, \rho_{\min}, F_3\}, & \text{if } F_2 \geq 0 \\ \min\{1, F_1\}, & \text{if } F_2 < 0 \end{cases} \tag{38}$$

$$F_0 = 1 - D_m(t)f_{m,local}/C_m(t) \tag{39}$$

$$F_1 = \left(D_m(t) - T_m^{mi}(t)\right)/\left(\frac{I_m(t)}{R_m^{Tr}(t)} + \frac{C_m(t)}{f_m(t)}\right) \tag{40}$$

$$F_2 = \alpha\left(\frac{I_m(t)}{R_m^{Tr}(t)} + \frac{C_m(t)}{f_m(t)}\right) - \beta\mathcal{K}C_m(t)f_{m,local}^2 + \beta\frac{P_mI_m(t)}{R_m^{Tr}(t)} \tag{41}$$

$$F_3 = \left(\frac{C_m(t)}{f_{m,local}} - T_m^{mi}(t)\right)/\left(\frac{C_m(t)}{f_{m,local}} + \frac{C_m(t)}{f_m(t)} + \frac{I_m(t)}{R_m^{Tr}(t)}\right) \tag{42}$$

$$F_4 = \beta\frac{P_mI_m(t)}{R_m^{Tr}(t)} - \beta\mathcal{K}C_m(t)f_{m,local}^2 - \alpha\frac{C_m(t)}{f_{m,local}} \tag{43}$$

$$
\begin{aligned}
f_1(m,t) = {} & \alpha(1-\rho_{m,1}(t))C_m(t)/f_{m,local} \\
& + \beta((1-\rho_{m,1}(t))\mathcal{K}C_m(t)f_{m,local}^2 \\
& + P_m\rho_{m,1}(t)I_m(t)/R_m^{Tr}(t))
\end{aligned} \tag{44}
$$

$$
\begin{aligned}
f_2(m,t) = {} & \alpha\left(\frac{\rho_{m,2}(t)I_m(t)}{R_m^{Tr}(t)} + \frac{\rho_{m,2}(t)C_m(t)}{f_m(t)} + T_m^{mi}(t)\right) \\
& + \beta((1-\rho_{m,2}(t))\mathcal{K}C_m(t)f_{m,local}^2 \\
& + P_m\rho_{m,2}(t)I_m(t)/R_m^{Tr}(t))
\end{aligned} \tag{45}
$$

Proof First, constraint $(C5)$ can be converted to:

$$\rho_m(t) \ge 1 - D_m(t)f_{m,local}/C_m(t) \tag{46}$$

$$\rho_m(t) \le (D_m(t) - T_m^{mi}(t))/\left(\frac{I_m(t)}{R_m^{Tr}(t)} + \frac{C_m(t)}{f_m(t)}\right) \tag{47}$$

Let

$$F_0 = 1 - D_m(t)f_{m,local}/C_m(t) \tag{39}$$

$$F_1 = (D_m(t) - T_m^{mi}(t))/\left(\frac{I_m(t)}{R_m^{Tr}(t)} + \frac{C_m(t)}{f_m(t)}\right) \tag{40}$$

When $F_0 > F_1$ or $F_1 < \rho_{\min}$, for any $\rho_m(t) \in [\rho_{\min}, 1]$, the minimum delay constraint cannot be satisfied, and problem $P2.1$ has no solution.

When $F_0 \le F_1$ and $F_1 \ge \rho_{\min}$, The proof of the optimal solution of problem $P2.1$ is:

When (1) and (2) in (35) are equal:

$$\rho_m(t) = \left(\frac{C_m(t)}{f_{m,local}} - T_m^{mi}(t)\right)/\left(\frac{C_m(t)}{f_{m,local}} + \frac{C_m(t)}{f_m(t)} + \frac{I_m(t)}{R_m^{Tr}(t)}\right) \tag{48}$$

Let

$$F_3 = \left(\frac{C_m(t)}{f_{m,local}} - T_m^{mi}(t)\right)/\left(\frac{C_m(t)}{f_{m,local}} + \frac{C_m(t)}{f_m(t)} + \frac{I_m(t)}{R_m^{Tr}(t)}\right) \tag{42}$$

Obviously, $0 < F_3 < 1$.

If $F_3 < \rho_{\min}$, then

$$\frac{(1-\rho_m(t))C_m(t)}{f_{m,local}} < \rho_m(t)\left(\frac{I_m(t)}{R_m^{Tr}(t)} + \frac{C_m(t)}{f_m(t)}\right) + T_m^{mi}(t) \tag{49}$$

Currently, let

$$
\begin{aligned}
F_5 = {} & \alpha\max\left\{\frac{(1-\rho_m(t))C_m(t)}{f_{m,local}}, \rho_m(t)\left(\frac{I_m(t)}{R_m^{Tr}(t)} + \frac{C_m(t)}{f_m(t)}\right) + T_m^{mi}(t)\right\} \\
& + \beta(\mathcal{K}(1-\rho_m(t))C_m(t)f_{m,local}^2 + P_m\rho_m(t)\frac{I_m(t)}{R_m^{Tr}(t)})
\end{aligned} \tag{50}
$$

Then

$$
\begin{aligned}
F_5 = {} & \rho_m(t)\left(\alpha\left(\frac{I_m(t)}{R_m^{Tr}(t)} + \frac{C_m(t)}{f_m(t)}\right)\right. \\
& \left. - \beta\mathcal{K}C_m(t)f_{m,local}^2 + \beta\frac{P_mI_m(t)}{R_m^{Tr}(t)}\right) \\
& + \beta\mathcal{K}C_m(t)f_{m,local}^2 + \alpha T_m^{mi}(t)
\end{aligned} \tag{51}
$$

(51) is a linear function of variable $\rho_m(t)$. Let

$$F_2 = \alpha\left(\frac{I_m(t)}{R_m^{Tr}(t)} + \frac{C_m(t)}{f_m(t)}\right) - \beta\mathcal{K}C_m(t)f_{m,local}^2 + \beta\frac{P_mI_m(t)}{R_m^{Tr}(t)} \tag{41}$$

When $F_2 \ge 0$, (51) is a monotonically increasing function of the variable $\rho_m(t)$. By combining (46), (47) and (51), the optimal offloading ratio is $\rho_m^*(t) = \max\{F_0, \rho_{\min}\}$. When $F_2 < 0$, (51) is a monotonic decreasing function of variable $\rho_m(t)$, and the optimal offloading ratio of user tasks is $\rho_m^*(t) = \min\{1, F_1\}$.

If $\rho_{\min} \le F_3 < 1$, (50) is a piecewise function. When $\rho_{\min} \le \rho_m(t) \le F_3$, (50) can be simplified as

$$
\begin{aligned}
F_5 = {} & \rho_m(t)\left(\beta\frac{P_mI_m(t)}{R_m^{Tr}(t)} - \beta\mathcal{K}C_m(t)f_{m,local}^2 - \alpha\frac{C_m(t)}{f_{m,local}}\right) \\
& + \alpha\frac{C_m(t)}{f_{m,local}} + \beta\mathcal{K}C_m(t)f_{m,local}^2
\end{aligned} \tag{52}
$$

When $F_3 \le \rho_m(t) \le 1$, (50) can be reduced to (51). (51) and (52) are linear functions of variable $\rho_m(t)$, let

$$F_4 = \beta\frac{P_mI_m(t)}{R_m^{Tr}(t)} - \beta\mathcal{K}C_m(t)f_{m,local}^2 - \alpha\frac{C_m(t)}{f_{m,local}} \tag{43}$$

The segment $\rho_{\min} \le \rho_m(t) \le F_3$ is called the first segment of the piecewise function. If $F_4 \ge 0$, then the first segment is a monotonically increasing function of $\rho_m(t)$, so the optimal solution of $\rho_m(t)$ is $\rho_{m,1}(t) = \max\{F_0, \rho_{\min}\}$. If $F_4 < 0$, then the first segment is a monotonic decreasing function, so the optimal solution of $\rho_m(t)$ is $\rho_{m,1}(t) = \min\{1, F_1, F_3\}$. The segment $F_3 \le \rho_m(t) \le 1$ is called the second segment of the piecewise function. If $F_2 \ge 0$, then the second segment is a monotonically increasing function of $\rho_m(t)$, so the optimal solution of $\rho_m(t)$ is $\rho_{m,2}(t) = \max\{F_0, \rho_{\min}, F_3\}$. If $F_2 < 0$, then the second segment is a monotonically decreasing function, so the optimal solution of $\rho_m(t)$ is $\rho_{m,2}(t) = \min\{1, F_1\}$.

Let

$$
\begin{aligned}
f_1(m,t) = {} & \alpha(1-\rho_{m,1}(t))C_m(t)/f_{m,local} \\
& + \beta((1-\rho_{m,1}(t))\mathcal{K}C_m(t)f_{m,local}^2 \\
& + P_m\rho_{m,1}(t)I_m(t)/R_m^{Tr}(t))
\end{aligned} \tag{44}
$$

$$f_2(m,t) = \alpha\left(\frac{\rho_{m,2}(t)I_m(t)}{R_m^{Tr}(t)} + \frac{\rho_{m,2}(t)C_m(t)}{f_m(t)} + T_m^{mi}(t)\right) + \beta((1-\rho_{m,2}(t))\mathcal{K}C_m(t)f_{m,local}^2 + P_m\rho_{m,2}(t)I_m(t)/R_m^{Tr}(t)) \quad (45)$$

The optimal value of problem $P2.1$ is the smaller of $f_1(m,t)$ and $f_2(m,t)$, so when $f_1(m,t) \le f_2(m,t)$, the optimal solution is $\rho_{m,1}(t)$, and $f_1(m,t) > f_2(m,t)$, the optimal solution is $\rho_{m,2}(t)$. Theorem 3 is proved.

4.2 BS selection and channel allocation

The outer problems related to user BS selection and channel allocation are:

$$\begin{aligned}
&P2.2: \min_{\boldsymbol{A}(t),\boldsymbol{B}(t)} B + \sum_{m\in\mathcal{M}} Q_m(t)S_m(t) + V\sum_{m\in\mathcal{M}} U_m(t)\\
&s.t.\,(C1)\sum_{n\in\mathcal{N}} a_m^n(t) = 1, \quad \forall m\in\mathcal{M}\\
&(C2)\sum_{l\in\mathcal{L}} b_m^l(t) = 1, \quad \forall m\in\mathcal{M}\\
&(C3)\sum_{m\in\mathcal{M}} a_m^n(t)b_m^l(t) = 1, \quad \forall n\in\mathcal{N},\, l\in\mathcal{L}\\
&(C5)T_m(t)\le D_m(t), \quad \forall m\in\mathcal{M}\\
&(C6)a_m^n(t), b_m^l(t)\in\{0,1\}, \quad \forall m\in\mathcal{M},\, n\in\mathcal{N},\, l\in\mathcal{L}
\end{aligned} \quad (53)$$

Problem $P2.2$ is a single time slot optimization problem. The current slot index t is omitted for the convenience of description. According to problem $P2.2$, this paper designs a joint optimization algorithm for task offloading ratio, user BS selection and channel allocation based on matching theory. The algorithm takes the user and the BS-channel pair (that is, one channel of a BS is a BS-channel pair) as the matching two-end elements to establish a one-to-one matching model [27]. According to the solution of the inner layer problem and problem $P2.2$, designs the preference value of one side element for the other side element.

In this paper, a channel of a BS is regarded as a whole, called a BS-channel pair, whose set is represented as $\mathcal{Y}$. In one-to-one matching, for any element $m\in\mathcal{M}$ (or $y\in\mathcal{Y}$) in user set (or BS-channel pair set), the matching object $y\in\mathcal{Y}$ (or $m\in\mathcal{M}$) is the element in the set on the other side, and the matching objects are unique and corresponding to each other (that is, if the matching object of m is y, the matching object of y is also m). In this paper, the one-to-one match between the current user and the BS-channel pair is represented by the character v. Obviously, $v\subset(\mathcal{M}\times\mathcal{Y})$.

The user BS selection and channel allocation obtained by one-to-one matching must meet the constraints $(C1)$, $(C2)$, $(C3)$, $(C6)$ of problem $P2.2$. To minimize the minimum value of the objective function of $P2.2$ and satisfy the constraint $(C5)$, this paper establishes a preference value of the user and the BS-channel pair for the matched object under the current matching to iteratively form the final matching result.

The preference value of user $m\in\mathcal{M}$ for the matched object under the current matching v is expressed as

$$\mathcal{P}_m(v) = indicator_{\{Ie=1\}} * (\eta - (B + Q_mS_m + VU_m)) \quad (56)$$

where, $Ie = 1$ indicates that the inner layer problem has a solution, that is, the maximum delay constraint can be satisfied. On the contrary, let $Ie = 0$, and η is a positive number far higher than the order of magnitude of the $P2.2$ objective function.

From the definition of preference value, when the matching object of user m under the current matching cannot finish the task within the maximum time delay, the user's preference value for the current matching v is 0. The smaller the average task cost, the higher the preference value of user m for the current matching object when the matching object of user m can make the task completion delay and MI meet the requirements.

The preference value of the BS-channel pair for the matched object under the current matching v is expressed as

$$\mathcal{P}_y(v) = \sum_{m\in\mathcal{M}} \mathcal{P}_m(v) \quad (57)$$

Because the interference between users is considered in this paper, the preference of users for their matching objects will be affected by the selection of other users. Therefore, the traditional stable matching may not exist. Therefore, the swap-matching method is used here to promote the iteration of the algorithm through the swap of user matching objects, so that the algorithm converges to a weak stable matching, that is, pairwise stability [28]. The concept of swap-blocking-pair used in the algorithm is explained here: in the current matching, if there are such two users, and their preference for their matching objects increases after they swap the matching objects, then these two users are called the current matching swap-blocking-pair. Because this paper considers the case that the product of the number of BSs and the number of channels is greater than the number of users, it is assumed that in the swap-matching, the BS-channel pair that is not matched with the user matches with the virtual user, and the preference value of the virtual user for its matching object is defined as 0.

The joint optimization algorithm for offloading ratio, BS selection and channel allocation based on matching theory is shown in algorithm 1:

Algorithm 1 Joint optimization algorithm of offloading ratio, BS selection and channel allocation

```
input: The user location, channel status, task size, queue backlog observed in the current time slot and the user BS connection in the previous time slot
output: Task offloading ratio, BS selection and channel allocation of the current time slot.
Initialization: Matched user set $Matched = \varnothing$, unmatched user set $unmatched = \mathcal{N}$, currently matched $v = \varnothing$, unmatched and BS-channel pair set $unmatchedp = \mathcal{Y}$, matched BS-channel pair set $matchedp = \varnothing$.
Initial match:
repeat
    for all Unmatched user $m \in unmatched$
        Calculate the preference value for all unmatched BS-channel pairs $y \in unmatchedp$ under the current matching $v$.
        Find the unmatched BS-channel pair with the highest preference value and send a matching request to it.
    end for
    for all Unmatched BS-channel pair $y \in unmatchedp$
        Calculate the preference value for each requesting user under the current matching $v$.
        Accept the request with the highest preference value, and both parties reach an initial match.
    end for
    Update the current matching $v$, matched user set $Matched$, unmatched user set $unmatched$, unmatched BS-channel pair set $unmatchedp$, matched BS-channel pair set $matchedp$.
until All users are matched
swap-matching stage:
    repeat
    set $v1 = v$
    for all users $m \in \mathcal{M}$
        for all BS-channel pairs $y \in \mathcal{Y}$
            if the user $m$ and user $m_y$ corresponding to BS-channel pair $y$ is a swap-blocking-pair in the current match.
                User $m$ and user $m_y$ swap matching objects.
                Update current match $v$.
        end for
    end for
until $v = v1$
return User BS selection, channel allocation and task offloading ratio in the current matching $v$
```

As shown in Algorithm 1, the input of the joint optimization algorithm of offloading ratio, BS selection and channel allocation is the observation value of the user position, channel status and task size of the current time slot. At the beginning of the algorithm, all users are in an unmatched state. Therefore, the matched user set *Matched* is an empty set. The unmatched user set *unmatched* is equal to the user set. The current matching v is an empty set. The unmatched BS-channel pair set *unmatchedp* is a BS-channel pair set. The matched BS-channel pair set *matchedp* is an empty set. The algorithm is divided into an initial matching stage and swap-matching stage. In the initial matching part, each user $m \in unmatched$ that has not yet matched the BS-channel pair will calculate the optimal offloading ratio when occupying each currently unmatched BS-channel pair $y \in unmatchedp$ according to the current matching v and calculate the preference value for each currently unmatched BS-channel pair according to the optimal offloading ratio. Finally, it will make a request for the BS-channel pair that has the highest preference value in the currently unmatched BS-channel pair. When all unmatched users make a request, each unmatched BS-channel pair will process the request it receives. It will find the request object with the highest preference value for itself under the current matching and match it. After all requests are processed, the set of matched users, matched BS-channel pairs, unmatched users, and unmatched BS-channel pairs will be updated. Then check whether there are unmatched users. If so, repeat the above steps until all users have a BS-channel pair to match. After the initial match is formed, users change the current match through swap operation until convergence. At the beginning of each round of swap in the swap phase, a matching $v1 = v$ before the current round of swap will be recorded, and each user $m \in \mathcal{M}$ will take turns to check whether it forms a swap-blocking-pair with other users m' in the currently matching (including virtual users). If satisfied, user m will swap matching objects with user m' and update the current match v. When the last user is checked, a new round of swaps will start again from the first user. The whole swap process lasts until there is no change in the matching before and after a round of swap operation, that is, $v1 = v$.

The convergence and paired stability of the proposed algorithm are demonstrated below.

Theorem 4 *Algorithm 1 will converge to a final match v_{fin} after finite iterations.*

Proof In each round of user requests in the initial matching phase, unmatched users will make requests to unmatched BS-channel pairs. In the worst case, all unmatched users submit requests to the same BS-channel pair, which can accept requests from one user at the request acceptance stage. Therefore, after a round of user requests and the BS accepts the request, at least one user will match successfully. After a certain round of request submission and reception, all users will match a BS-channel pair, and the initial matching phase will end. In the swap phase, it can be seen from the meaning of the swap-blocking-pair that each swap of user matching objects will lead to a strict decrease in the cost of the entire system. Because the cost function of the entire system is a bounded function, the swap phase will terminate after a certain number of swaps by users.

Theorem 5 *Algorithm 1 will eventually converge to a pairwise-stable match.*

Proof First, the concept of pairwise stability is explained: if there is no user who satisfies the definition of swap-blocking-pair in a match, the match is pairwise stable. Here, we prove the pairwise-stability of the final match by contradiction. If the final match v_{fin} is not pairwise-stable, then there is at least one pair of users m, m' in the match, which is recorded as the swap-blocking pair of the current match. According to the steps of algorithm 1, the algorithm will swap the BS-channel pairs matched by the user m and user m' in the swap phase, and then update the matching. This contradicts the fact that matching v_{fin} is the final match, so the final match is pairwise-stable.

It should be clarified here that although this paper uses Lyapunov theory to transform the original optimization problem and proves that the optimal solution of the original problem can be obtained by solving the transformed problem with control parameters $V \rightarrow \infty$, the matching-based two-level optimization algorithm used in this paper is used to solve the transformed problem. The algorithm is a heuristic algorithm, and the solution it obtains does not reach the optimum of the original problem.

Finally, the complexity of Algorithm 1 is analyzed.

In the initial matching phase of the algorithm, the worst case is that only one user completes the initial matching in each round of request making and receiving, so the request making and receiving phase can be performed up to M times. At each request submission stage, each unmatched user needs to change the current match and calculate the preference value for the BS- channel pair matched after changing the match. Therefore, the cycle in the request submission stage can be performed at most $\sum_{i=0}^{M-1} (M-i)(NL-i)$ times. For each request acceptance phase, each unmatched BS-channel pair needs to compare the preference value of the received request and select the matched object, so this phase can be cycled at most $\sum_{i=0}^{M-1} (NL-i)$ times. In each round of swap in the matching swap stage, the algorithm should check whether there are users constituting the current matching swap-blocking-pair. Assuming that there is a total of J rounds in the swap process, the number of cycles in the swap stage is $JMNL$. Therefore, the complexity of the whole Algorithm 1 is $O\left(\sum_{i=0}^{M-1} (M-i)(NL-i) + \sum_{i=0}^{M-1}(NL-i) + MJNL)\right)$.

Since the transformed problem is a single time slot problem, it is required that the algorithm has to be able to complete this within the required time duration. From the obtained time complexity, it can be seen that the execution time of the proposed algorithm increases with the increase of the input size of the algorithm (number of base stations, users, channels). When the input size (number of base stations, users and channels) is small, the proposed algorithm can meet the corresponding time requirements, but when the input size is too large, the algorithm may not be able to complete in one time slot. Therefore, a lower time complexity algorithm may be able to facilitate the implementation of the algorithm. We will be looking at less complex solutions in future work.

5 Simulation and performance analysis

The proposed algorithm is simulated by MATLAB. In the simulation, 5 BSs are deployed in the center and four corners of a 600×600 m^2 square area. The user's movement trajectory is generated by a random walk model [29] and lasts for 1000 time slots, with each time slot length of $\tau = 1s$. The path loss model is $pathloss_{m,n} = 35.3 + 37.6\log_{10}(d_{m,n})$, where $d_{m,n}$ denotes the distance between user m and target n. Small-scale fading model is represented by Rayleigh distribution random variable of unit mean. The target frequency response in the sensing model is assumed to adhere to the standard normal distribution [19]. The number of integrated waveform symbols used is $K = 10$, the symbol duration is $T_s = 5$ µs, and the minimum value of MI is $MI_{\min} = 80$ bit. The amount of task data generated by users in each time slot is subject to uniform distribution $[1, 1.5] \times 10^6$ bit, task intensity is $r_m = 737.5$ cycles/bit, and the maximum delay requirement of the task is $D_m(t) = 1$ s. The user's sensing failure rate is required to be $\delta = 0.01$, and the minimum unloading ratio is required to be $\theta_{\min} = 0.01$. Other simulation parameters are shown in Table 1.

Three benchmark algorithms are compared with algorithm 1. The first benchmark algorithm is the algorithm in [4]. This algorithm considers the static scenario in which the user's task is fully offloaded and optimizes the user's BS selection and channel allocation. The second benchmark algorithm does not consider the optimization of user BS selection. Its BS selection adopts the scheme of nearby access. The third benchmark algorithm does not support user service migration, that is, the user always keeps the original BS selection unchanged. When the task completion delay exceeds the maximum delay requirement or the MI does not meet the minimum requirement, this paper considers the task to be failed, and treats the task whose completion time exceeds the maximum delay limit ($D_m = 1$ s) as discarded, that is, the task completion time is

Table 1 Simulation parameters

Parameter	Symbol	Value
Number of users (default)	M	20
Number of BSs	N	5
Number of channels	L	10
User transmission power	P_m	23 dBm
Channel bandwidth	W	0.5 MHz
Noise power spectral density	N_0	− 174 dBm
User calculated power factor	$\mathcal{K}$	$10^{(-28)}$
User computing capacity	$f_{m,local}$	1 GHz
BS computing capacity	F_n	30 GHz
User delay weight coefficient	α	0.5
Weight coefficient of user energy consumption	β	5
User migration cost	ε	0.1
Control parameters	V	40

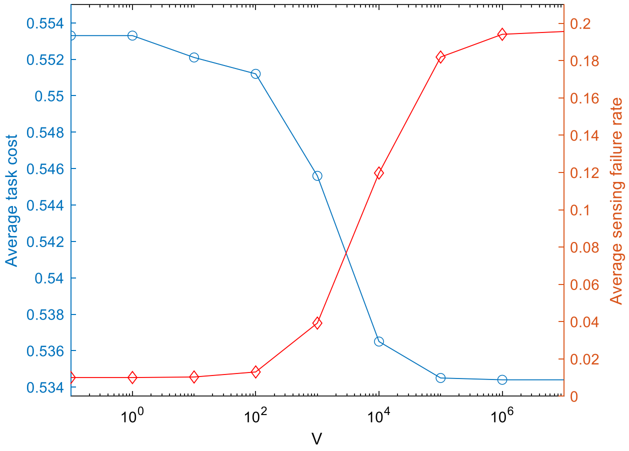
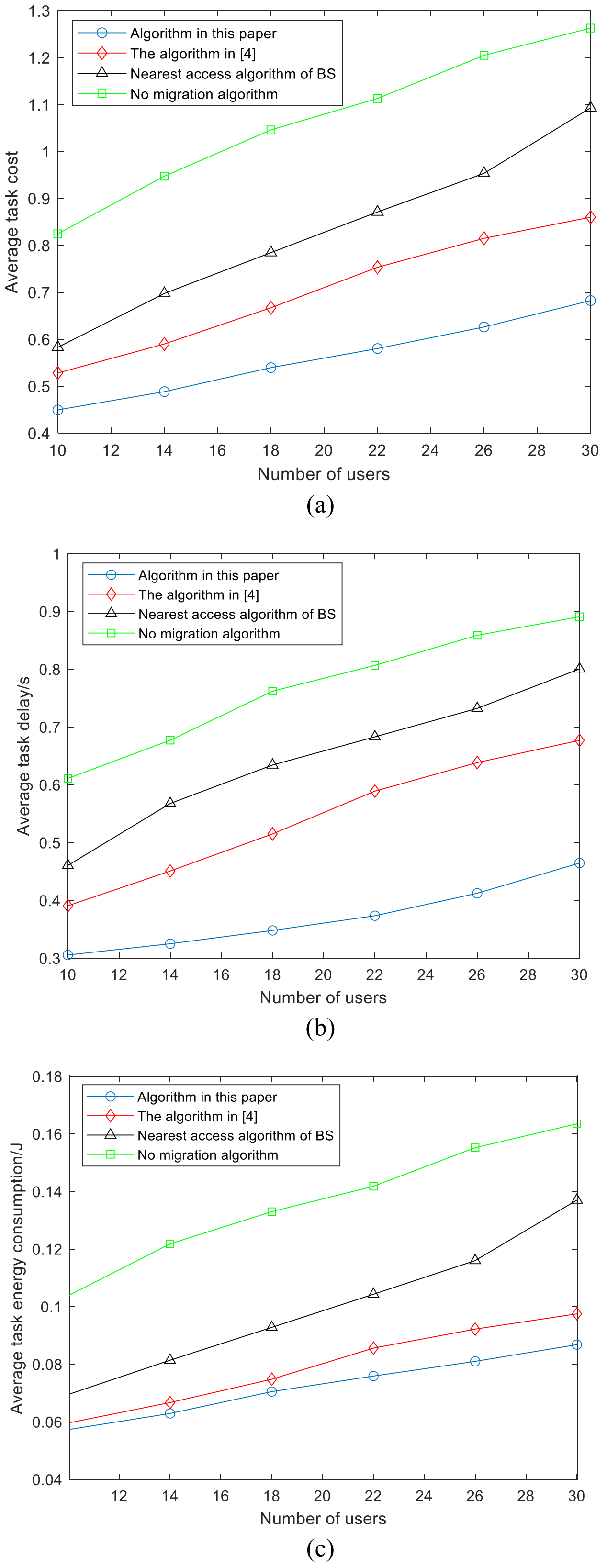


Fig. 2 Effects of control parameter V on average task cost and sensing failure rate

counted according to the maximum delay limit D_m, and the task data transmission energy consumption is counted according to $P_m * D_m$.

As can be seen from Fig. 2, the average task cost gradually decreases and tends to be optimal as the control parameter V gradually increases. The sensing failure rate gradually decreases with the increase of V. Figure 2 clearly demonstrates that the control parameter V can equalize the average task cost and perceived failure rate.

Figure 3(a–d) shows the effects of the number of users on performance (average task cost, average task energy consumption, average task delay and average task failure rate) under different schemes. The figures show that the average task cost, delay, energy consumption and failure rate all rise as the number of users increases. The main reason is that (1) the total amount of computing resources of the whole system is limited and constant. As the number of users rises, the computing resources allotted to each user will be reduced, and the computing delay of the task will gradually increase. (2) The increase of the number of users will also increase the probability of different BS users multiplexing the same channel. This enhances the interference between users and reduces the user's data transmission rate, thus increasing the user's task transmission delay and energy consumption. (3) The enhancement of interference between users will also reduce the user's MI. The reduction of MI will urge users to choose a channel with better sensing performance. However, this channel may have a large communication fading, further increasing the user's transmission delay and energy consumption. (4) The shortage of resources on the BS side will lead to more tasks being executed locally, which will increase the task completion time and energy consumption. (5) The increase of task completion delay and the decrease of MI will increase the probability of task timeout or failure to meet the minimum MI requirements. In addition, algorithm 1 outperforms the other three schemes. From Fig. 3(a–c), it can be shown that the average task cost, average task delay and energy consumption of the algorithm in this paper are superior to those obtained by the algorithm in reference [4]. The reason is that the algorithm in [4] adopts the strategy of full task offloading, while the algorithm in this paper considers the optimization of partial offloading and task offloading ratio, which enables the user task to use local computing resources and BS server computing resources in the meantime, and the terminal and BS server can process tasks in parallel. It reduces the task completion time and transmission energy consumption. In addition, the algorithm in [4] does not consider user mobility, while the algorithm in this paper considers the migration cost brought by user mobility when optimizing, so that better BSs can be selected for access. It can be demonstrated from Fig. 2(d) that when there are few users, the algorithm used in this paper has a greater task failure rate than the algorithm in [4]. This is because this paper considers that the

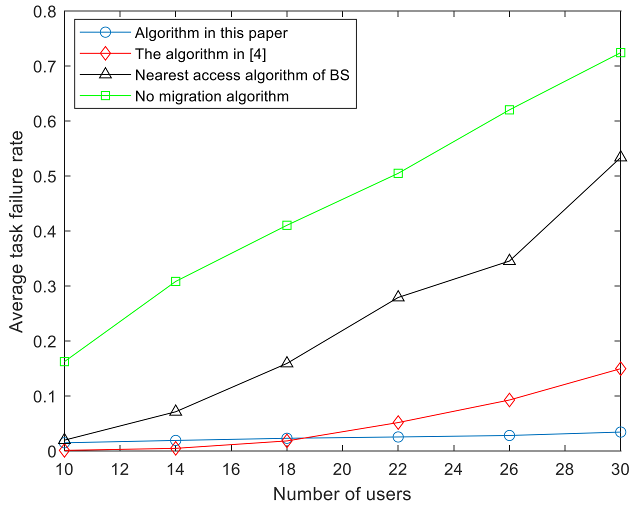


(a)

(b)

(c)

Fig. 3 Effects of the number of users under different schemes **a** Average task cost **b** Average task delay **c** Average task energy consumption **d** Average task failure rate



(d)

Fig. 3 continued

long-term sensing failure rate of users is required to reach a certain threshold requirement, that is, certain task sensing failure is allowed, while the algorithm in [4] requires that the MI of users in each time slot should meet the minimum requirement. When the number of users increases, the algorithm in this paper has a lower failure rate what is found in [4]. This is because the optimization of the task offloading ratio reduces the task completion delay, thus reducing the possibility of task timeout. At this time, the task can choose a channel that is more conducive to the sensing performance, so that the sensing failure rate is reduced.

Figure 4(a–d) shows the effects of average task data volume on performance under different schemes. The figures show that the average task cost, delay, energy consumption, and failure rate increase as the average task data volume increases. The reason is that when the average task data volume increases, the transmission and calculation time of the task offloaded to the BS will increase, leading to the decrease of the user task offloading ratio, and the increase of the average task completion delay and energy consumption due to more tasks being executed locally. The probability of the task exceeding the maximum delay requirement will also increase. From Fig. 4(a–c), it can be demonstrated that the average task cost, delay, and energy consumption of the algorithm in this paper are superior to that of the algorithm in [4]. The reason is like the above. The algorithm in this paper considers the service migration cost caused by user mobility when optimizing the algorithm, and the partial offloading of tasks and the optimization of the offloading ratio makes the tasks processed in parallel locally and remotely, reducing the user completion delay and energy consumption, thus reducing the

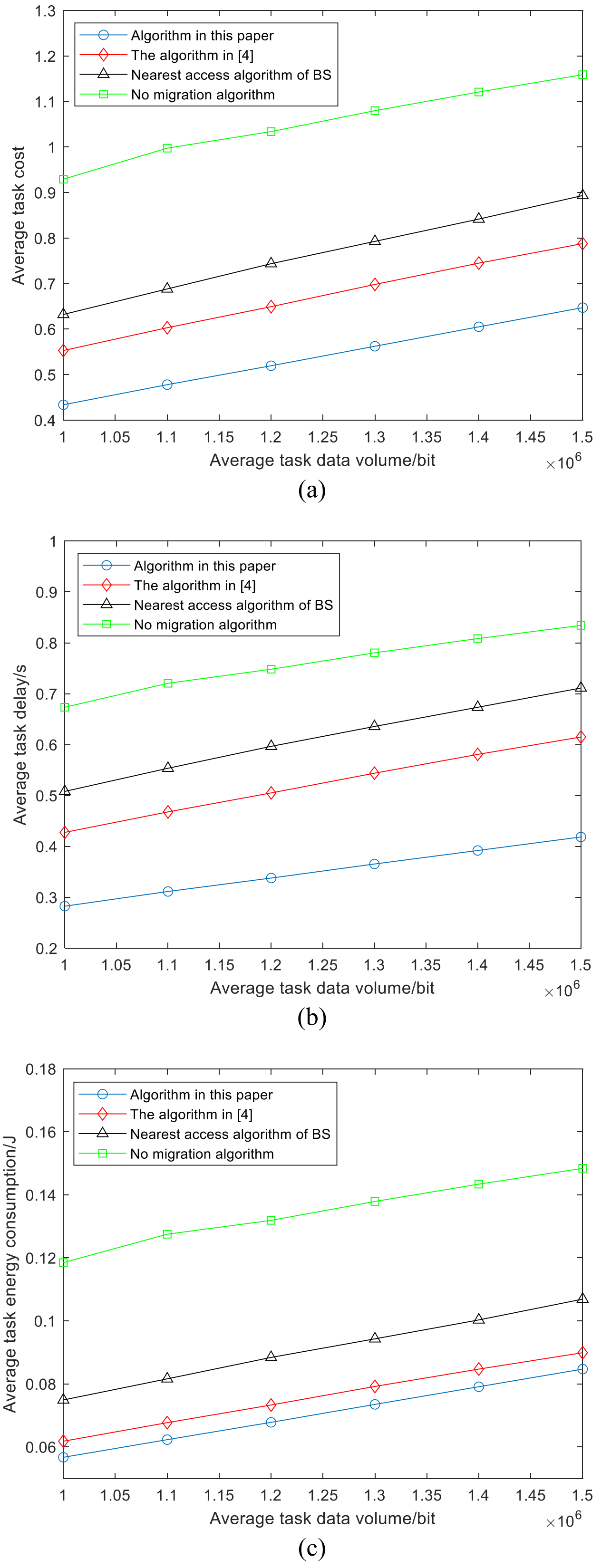


Fig. 4 Effects of average task data volume under different schemes **a** Average task cost **b** Average task delay **c** Average task energy **d** Average task failure rate

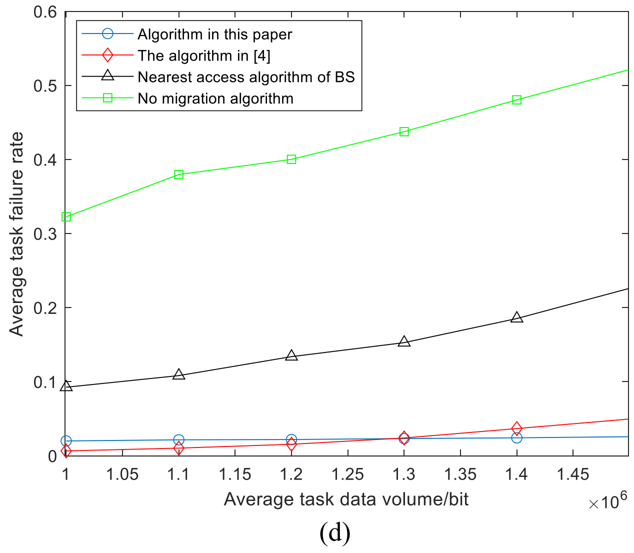
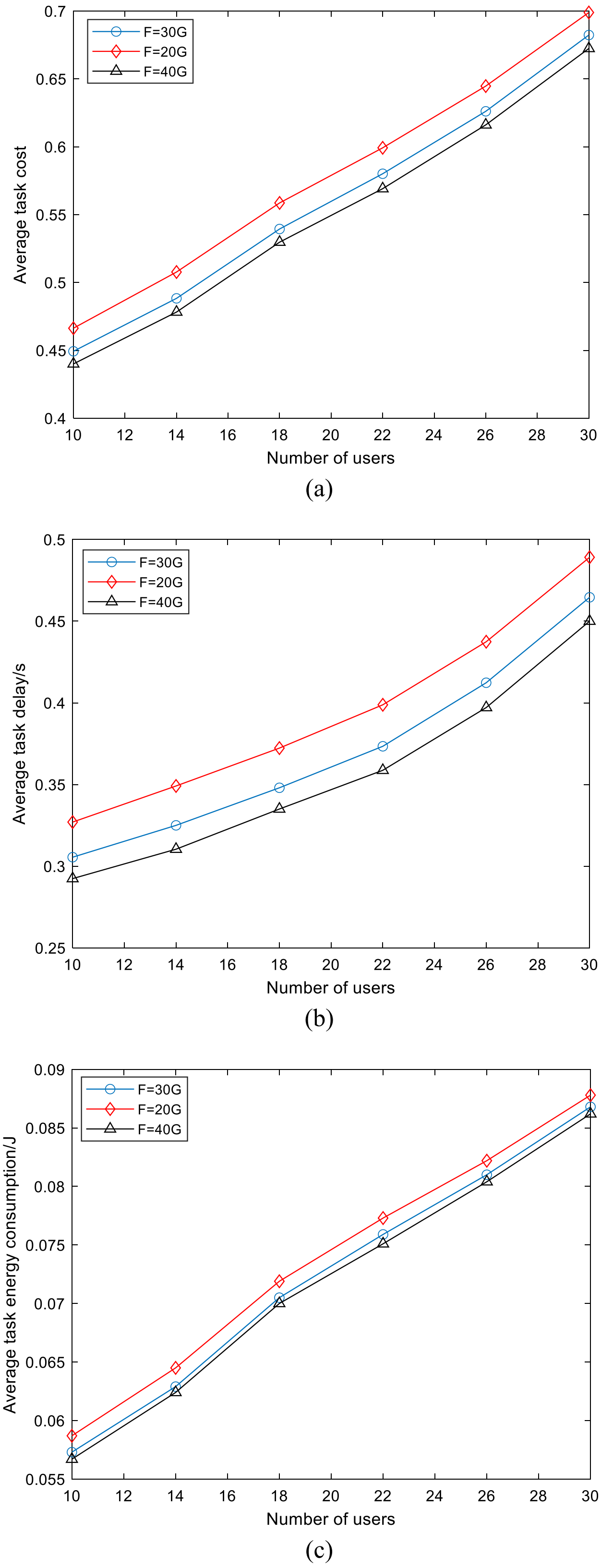


Fig. 4 continued

average task cost. From Fig. 4(d), the average task failure rate is higher than that in [4] when the average task data volume is small. This is because this paper considers that the sensing failure rate of users in multiple timeslots reaches a certain threshold, which allows the certain sensing failures. When the amount of task data becomes larger, the algorithm in this paper compared with the one in [4] has a decreased failure rate. This is because the increase in average task data volume makes more tasks unable to be completed within the maximum delay requirements, resulting in a gradual increase in the task failure rate. However, the partial offloading of tasks and the optimization of the offloading ratio reduces the user's completion delay, and leads to a reduction in the task failure rate.

Figure 5(a–d) shows the effects of different BS computing resources on performance. It can be seen from Fig. 5(a–c) that with the decrease of computing resources of the BS, the average task cost, delay, and energy consumption increase. The main reason is that when the computing resources of the BS become less, the computing resources that the BS allotted to each task become less, which causes an increase in the computing delay of the task offloaded to the BS. Therefore, the tasks are more calculated locally, which increases the task completion delay and energy consumption, Finally, the average task cost increases. Figure 5(d) presents that the reduction of task computing resources may lead to a slight decrease in the average task failure rate. This is because the reduction of the BS computing resources will bring a decrease in task offloading ratio, resulting in a reduction in the amount of data transmitted by the task to the BS. The reduction of the amount of data transmitted by the task makes the decrease

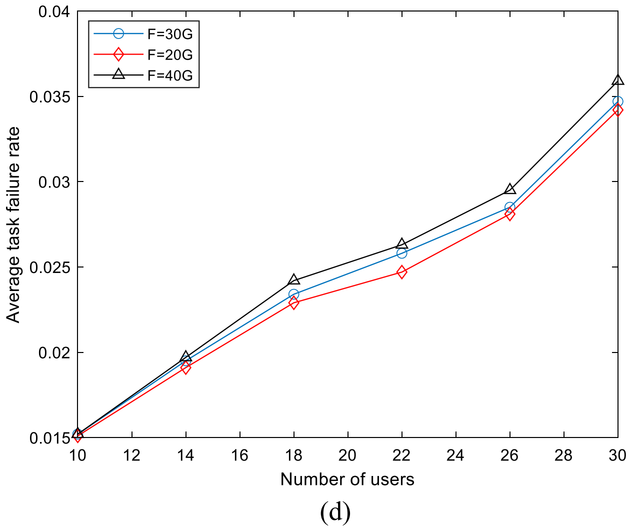


(a)

(b)

(c)

Fig. 5 Effects of computing resources of different BSs **a** Average task cost **b** Average task delay **c** Average task energy consumption **d** Average task failure rate

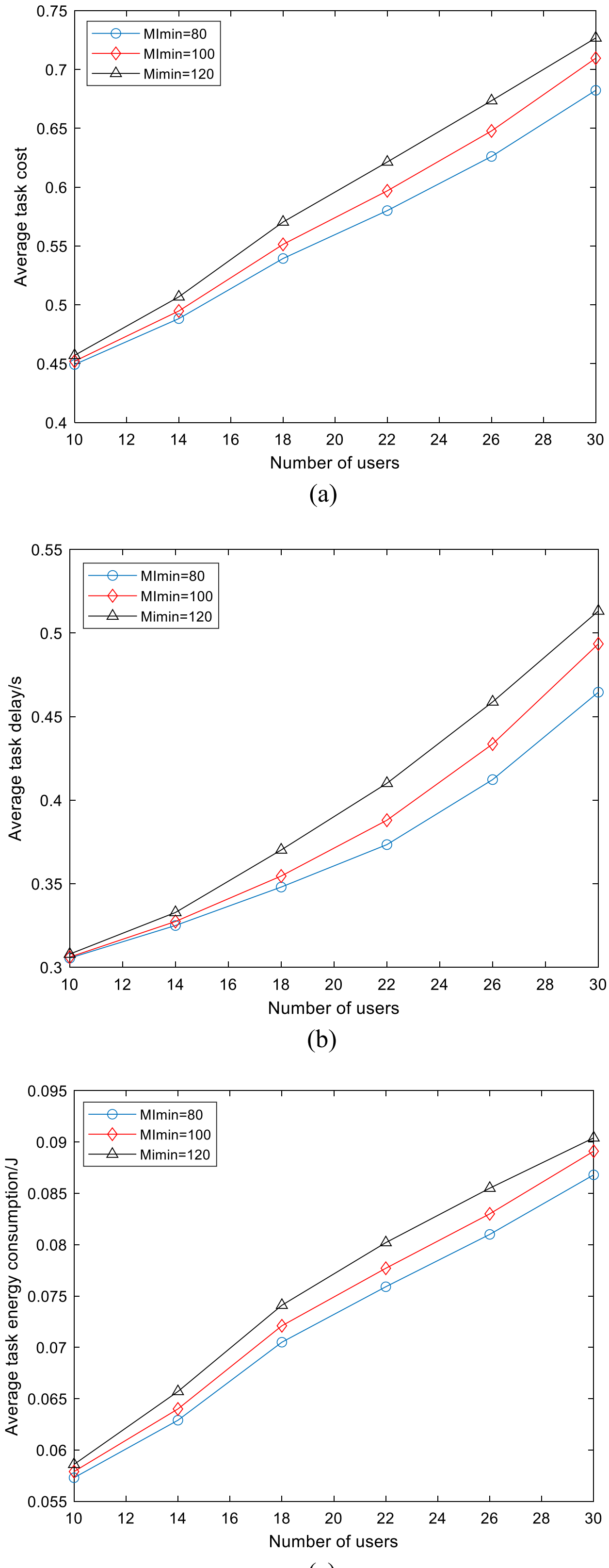


(d)

Fig. 5 continued

of the task completion delay and energy consumption in the channel with better communication performance less than the increase of the sensing failure rate in the channel with better sensing performance. Therefore, users tend to occupy the channel with good sensing performance, resulting in a reduction of task sensing failure. Because the algorithm in this paper considers the partial offloading of tasks and the optimization of the offloading ratio, the task delay requirement is relaxed and the tasks exceeding the maximum delay requirement will not increase. In summary, the average task failure rate will decrease.

Figure 6(a–d) shows the effects of different $MI_{\min}$ (minimum MI) requirements on performance. It can be demonstrated from the figure that the improvement of the requirement will increase the average task cost, delay, energy consumption and failure rate. Because the improvement of the requirement will make the user more inclined to choose the channel with good sensing performance. Such channels may have a large communication fading, resulting in a reduction in data transmission rate, an increase in task transmission time and energy consumption. More tasks are calculated locally, resulting in an increase in the total task completion time and energy consumption. The improvement of $MI_{\min}$ also makes more tasks judged as sensing failure, so the task failure rate also increases. It can also be presented from the figure that as the number of users increases, the effects of improving $MI_{\min}$ on performance will become more and more serious. The reason is that when the number of users is large, the interference between users will become increasingly serious, and it will be more difficult to sense MI to meet the minimum requirements. Therefore, the sensing failure rate will increase sharply. Since the requirements of sensing failure rate should be met as far as possible, users are more

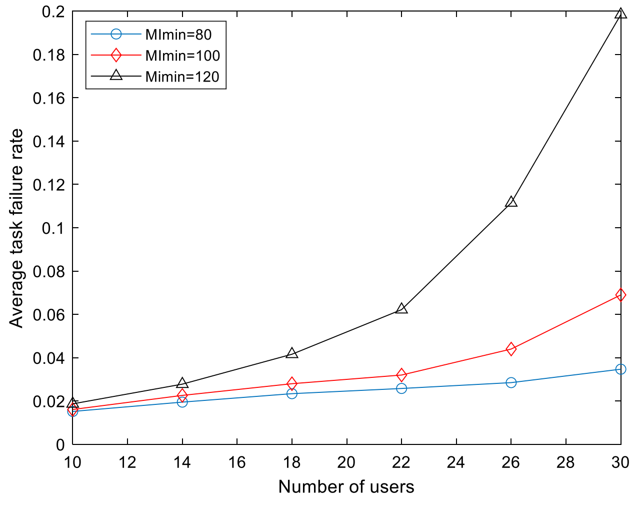


(a)

(b)

(c)

Fig. 6 Effects of different $MI_{\min}$ requirements **a** Average task cost **b** Average task delay **c** Average task energy consumption **d** Average task failure rate

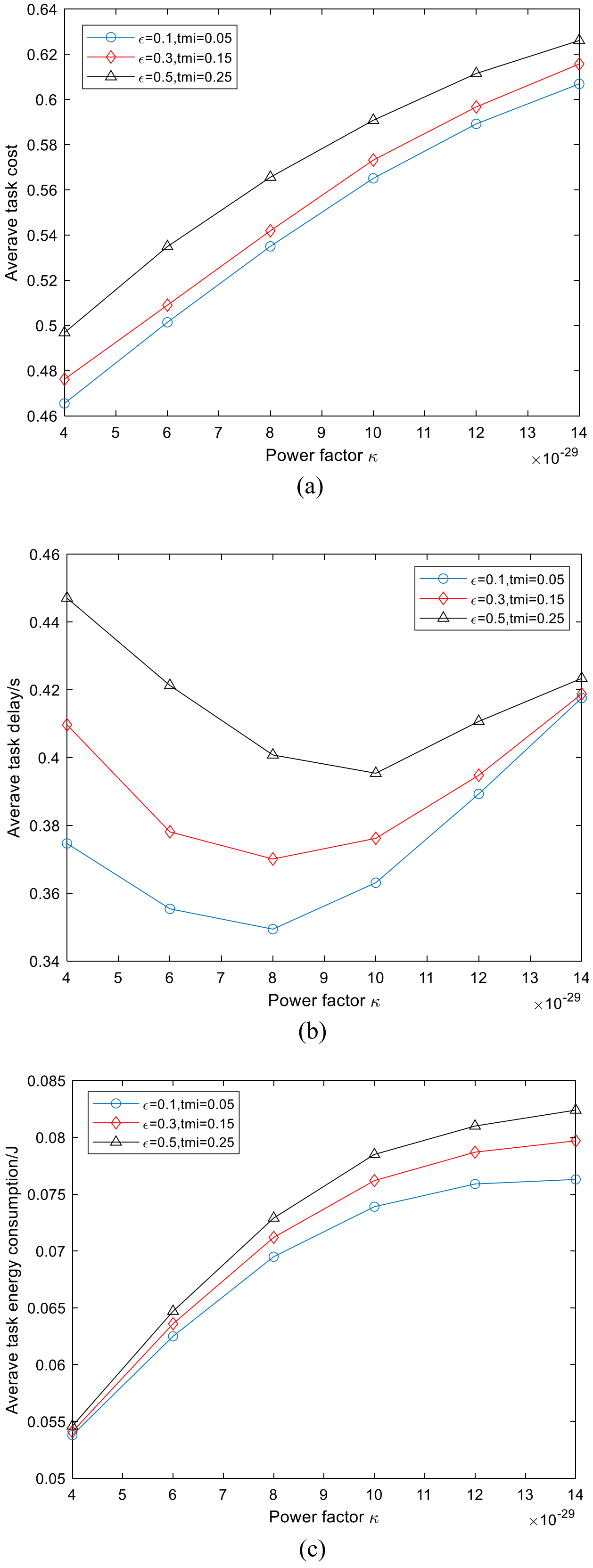


(d)

Fig. 6 continued

inclined to choose channels with good sensing performance. Thus, the average task cost, delay and energy consumption increase more rapidly.

Figure 7(a–d) show the effects of power factor and migration cost on average task cost, delay, and energy consumption. The horizontal and vertical coordinates of each line type show the effect of different power factors on the analysed metrics, and different line types show the effect of different migration costs on the analysed metrics. In the legend, ε shows the resource consumption of the migration and t_{mi} shows the delay consumption of the migration. From Fig. 7(a), it can be seen that the average task cost increases with the increase of power factor. This is because the increase in power factor increases the energy consumption of local computation. From Fig. 7(b), it can be seen that the latency decreases and then increases when the power factor increases. This is because too large or too small a power factor makes the remote processing latency greater than the local processing latency or the local processing latency greater than the remote processing latency, resulting in the time parallelization between remote and local processing not being better utilized. From Fig. 7(c), it can be seen that the energy consumption gradually increases when the power factor increases. This is because the increase in the power factor increases the energy consumption of the local computation. From Fig. 7(d), it can be seen that the task failure rate increases when the power factor increases. This is because the rise in task cost makes the proposed algorithm pay more attention to the optimization of task cost, and the optimization of partial offloading makes the task failure due to sensing failure dominate, which eventually increases the task failure rate. From Fig. 7(a–d), it can be seen that the average task failure rate, delay, energy consumption and task failure rate

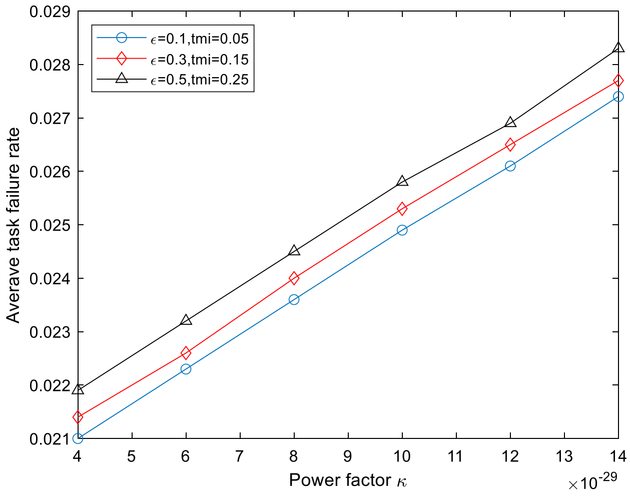


(a)

(b)

(c)

Fig. 7 Effects of different migration costs and power **a** Average task cost **b** Average task delay **c** Average task energy consumption **d** Average task failure rate



(d)

Fig. 7 continued

when the migration cost rises. all rise. This is because the increase in migration cost makes users more inclined to keep the original base station connection, which prevents them from choosing the optimal base station and channel.

6 Conclusion

In the ISCC scenario of multi-user and multi-BS, this paper considers the user mobility and partial offloading, and establishes a problem of minimizing the sum of the user's long-term cost. The problem considers the user's latency, energy consumption, and migration costs as well as the sensing. The problem is converted to a single time slot problem according to Lyapunov optimization theory, and the matching method is used to solve the converted problem. Simulation results reveal that the proposed algorithm outperforms other benchmark algorithms. Because the flexible location of mobile edge nodes can further improve the quality of service for users and the generation of multiple tasks per user in a single time slot can fit a wider range of realistic scenarios, future work may consider resource allocation and offloading decisions in ISCC scenarios with multitasking users and mobile edge nodes.

Acknowledgement This work is supported by the National Natural Science Foundation of China (61971239, 92067201), Jiangsu Provincial Key Research and Development Program (No. BE2022068-2).

References

1. Shirin Abkenar, F., Ramezani, P., Iranmanesh, S., Murali, S., Chulerttiyawong, D., Wan, X. Y., Jamalipour, A., & Raad, R.

(2022). A survey on mobility of edge computing networks in IoT: State-of-the-Art, architectures, and challenges. *IEEE Communications Surveys & Tutorials, 24*(4), 2329–2365. https://doi.org/10.1109/COMST.2022.3211462

2. Liu, A., Huang, Z., Li, M., Wan, Y. B., Li, W. R., Han, T. X., Liu, C. C., Du, R., Tan, D. K. P., Lu, J. M., Shen, Y., Colone, F., & Chetty, K. (2022). A survey on fundamental limits of integrated sensing and communication. *IEEE Communications Surveys & Tutorials, 24*(2), 994–1034. https://doi.org/10.1109/COMST.2022.3149272
3. Feng, Z., Wei, Z., Chen, X., Yang, H., Zhang, Q., & Zhang, P. (2021). Joint communication, sensing, and computation enabled 6G intelligent machine system. *IEEE Network, 35*(6), 34–42. https://doi.org/10.1109/MNET.121.2100320
4. Zhao, L., Wu, D., Zhou, L., & Qian, Y. (2022). Radio resource allocation for integrated sensing, communication, and computation networks. *IEEE Transactions on Wireless Communications, 21*(10), 8675–8687. https://doi.org/10.1109/TWC.2022.3168348
5. Kao, Y.-H., Krishnamachari, B., Ra, M.-R., & Bai, F. (2017). Hermes: Latency optimal task assignment for resource-constrained mobile computing. *IEEE Transactions on Mobile Computing, 16*(11), 3056–3069. https://doi.org/10.1109/TMC.2017.2679712
6. Wang, C., Yu, F. R., Liang, C., Chen, Q., & Tang, L. (2017). Joint computation offloading and interference management in wireless cellular networks with mobile edge computing. *IEEE Transactions on Vehicular Technology, 66*(8), 7432–7445. https://doi.org/10.1109/TVT.2017.2672701
7. Bi, J., Yuan, H., Duanmu, S., Zhou, M., & Abusorrah, A. (2021). Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization. *IEEE Internet of Things Journal, 8*(5), 3774–3785. https://doi.org/10.1109/JIOT.2020.3024223
8. Wu, Y.-C., Dinh, T. Q., Fu, Y., Lin, C., & Quek, T. Q. S. (2021). A hybrid DQN and optimization approach for strategy and resource allocation in MEC networks. *IEEE Transactions on Wireless Communications, 20*(7), 4282–4295. https://doi.org/10.1109/TWC.2021.3057882
9. Yuan, H., Guo, D., Tang, G., & Luo, L. (2022). Online energy-aware task dispatching with QoS guarantee in edge computing. *Chinese Journal on Internet of Things., 5*(2), 71–77. https://doi.org/10.11959/j.issn.2096-3750.2021.00230
10. Peng, J., Qiu, H., Cai, J., Xu, W., & Wang, J. (2021). D2D-assisted multi-user cooperative partial offloading, transmission scheduling and computation allocating for MEC. *IEEE Transactions on Wireless Communications, 20*(8), 4858–4873. https://doi.org/10.1109/TWC.2021.3062616
11. Saleem, U., Liu, Y., Jangsher, S., Tao, X., & Li, Y. (2020). Latency minimization for D2D-enabled partial computation offloading in mobile edge computing. *IEEE Transactions on Vehicular Technology, 69*(4), 4472–4486. https://doi.org/10.1109/TVT.2020.2978027
12. Zhan, W., Luo, C., Min, G., Wang, C., Zhu, Q., & Duan, H. (2022). Mobility-aware multi-user offloading optimization for mobile edge computing. *IEEE Transactions on Vehicular Technology, 69*(3), 3341–3356. https://doi.org/10.1109/TVT.2020.2966500
13. Yang, G., Hou, L., He, X., He, D., Chan, S., & Guizani, M. (2021). Offloading time optimization via markov decision process in mobile-edge computing. *IEEE Internet of Things Journal, 8*(4), 2483–2493. https://doi.org/10.1109/JIOT.2020.3033285
14. Liang, Z., Liu, Y., Lok, T.-M., & Huang, K. (2022). A two-timescale approach to mobility management for multicell mobile edge computing. *IEEE Transactions on Wireless Communications, 21*(12), 10981–10995. https://doi.org/10.1109/TWC.2022.3188695
15. Labriji, I., Meneghello, F., Cecchinato, D., Sesia, S., Perraud, E., Calvanese Strinati, E., & Rossi, M. (2021). Mobility aware and dynamic migration of MEC services for the internet of vehicles. *IEEE Transactions on Network and Service Management, 18*(1), 570–584. https://doi.org/10.1109/TNSM.2021.3052808
16. Saleem, U., Liu, Y., Jangsher, S., Li, Y., & Jiang, T. (2021). Mobility-aware joint task scheduling and resource allocation for cooperative mobile edge computing. *IEEE Transactions on Wireless Communications, 20*(1), 360–374. https://doi.org/10.1109/TWC.2020.3024538
17. Hu, H., Wang, Q., Hu, R. Q., & Zhu, H. (2021). Mobility-aware offloading and resource allocation in a MEC-enabled IoT network with energy harvesting. *IEEE Internet of Things Journal, 8*(24), 17541–17556. https://doi.org/10.1109/JIOT.2021.3081983
18. Nguyen, H. T., Hoang, D. T., Luong, N. C., Niyato, D., & Kim, D. I. (2021). A hierarchical game model for OFDM integrated radar and communication systems. *IEEE Transactions on Vehicular Technology, 70*(5), 2077–2082. https://doi.org/10.1109/TVT.2021.3069431
19. Liu, Y., Liao, G., Xu, J., Yang, Z., & Zhang, Y. (2017). Adaptive OFDM integrated radar and communications waveform design based on information theory. *IEEE Communications Letters, 21*(10), 2174–2177. https://doi.org/10.1109/LCOMM.2017.2723890
20. Shi, C., Wang, Y., Wang, F., Salous, S., & Zhou, J. (2021). Joint optimization scheme for subcarrier selection and power allocation in multicarrier dual-function radar-communication system. *IEEE Systems Journal, 15*(1), 947–958. https://doi.org/10.1109/JSYST.2020.2984637
21. Qi, Q., Chen, X., Khalili, A., Zhong, C., Zhang, Z., & Ng, D. W. K. (2022). Integrating sensing, computing, and communication in 6G wireless networks: Design and optimization. *IEEE Transactions on Communications, 70*(9), 6212–6227. https://doi.org/10.1109/TCOMM.2022.3190363
22. Ding, Z., Xu, D., Schober, R., & Poor, H. V. (2022). Hybrid NOMA offloading in multi-user MEC networks. *IEEE Transactions on Wireless Communications, 21*(7), 5377–5391. https://doi.org/10.1109/TWC.2021.3139932
23. Chai, R., Lin, J., Chen, M., & Chen, Q. (2019). Task execution cost minimization-based joint computation offloading and resource allocation for cellular D2D MEC systems. *IEEE Systems Journal, 13*(4), 4110–4121. https://doi.org/10.1109/JSYST.2019.2921115
24. Liang, Z., Liu, Y., Lok, T.-M., & Huang, K. (2021). Multi-cell mobile edge computing: joint service migration and resource allocation. *IEEE Transactions on Wireless Communications, 20*(9), 5898–5912. https://doi.org/10.1109/TWC.2021.3070974
25. Mao, Y., Zhang, J., Song, S. H., & Letaief, K. B. (2017). Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems. *IEEE Transactions on Wireless Communications, 16*(9), 5994–6009. https://doi.org/10.1109/TWC.2017.2717986
26. Neely, M. (2010). *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool
27. Roth, A. E., & Sotomayor, M. A. O. (1992). *Two-Sided Matching: A Study In Game-Theoretic Modeling and Analysis*. Cambridge Univ. Press.
28. Bodine-Baron, E., Lee, C., Chong, A., Hassibi, B., & Wierman, A. (2011). Peer effects and stability in matching markets. In eds, (Ed.), *Persiano, G* (pp. 117–129). Algorithmic Game Theory. https://doi.org/10.1007/978-3-642-24829-0_12

29. Sun, Y., Zhou, S., & Xu, J. (2017). EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks. *IEEE Journal on Selected Areas in Communications, 35*(11), 2637–2646. https://doi.org/10.1109/JSAC.2017.2760160

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Shuo Sun is currently pursuing the master's degree in telecommunication engineering with the Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests are in the area of mobile edge computing.

Qi Zhu received the bachelor's and master's degrees in radio engineering from the Nanjing University of Posts and Telecommunications (NUPT), Nanjing, China, in 1986 and 1989, respectively. She is currently a Professor with the School of Telecommunication and Information Engineering, NUPT. Her research interests include wireless networks, Internet of Things, green communication, and mobile edge computing.