EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Joint computation offloading and resource allocation strategy for D2D-assisted and NOMA-empowered MEC systems

Umar Ajaib Khan[1*], Rong Chai[1], Shabeer Ahmad[2] and Waleeed Almughalles[1]

*Correspondence: umarajaibkhan@gmail.com

[1] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China
[2] School of Electronics Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China

Abstract

Multi-access edge computing (MEC) emerged as a promising network paradigm that provides computation, storage and networking features within the edge of the pervasive mobile radio access network. This paper jointly considers computation offloading and resource allocation problem in device-to-device (D2D)-assisted and non-orthogonal multiple access (NOMA)-empowered MEC systems, where each mobile device (MD) is allowed to execute its task in one of the three ways, i.e., local computing, MEC offloading or D2D offloading. We invoke orthogonal multiple access (OMA) and NOMA schemes for MDs that select D2D offloading mode, allowing them to assign tasks to their peers using OMA or NOMA. The original problem is formulated as an overall energy consumption minimization problem, which proves to be NP-hard, making it intractable to solve optimally. We start from a simple case, OMA case and transform the original problem into two sub-problems, i.e., resource allocation sub-problem and computation offloading sub-problem and propose two heuristic algorithms to obtain the sub-optimal solutions of both sub-problems. Then, for the MDs selecting D2D offloading mode, we conduct user pairing and apply the NOMA scheme. Finally, simulation results demonstrate the efficiency of the proposed scheme when compared with the related schemes.

Keywords: Multi-access edge computing, Computation offloading, Resource allocation, Non-orthogonal multiple access, Delay, Energy consumption, Energy efficiency

1 Introduction

Technology scaling triggers some promising applications, i.e., face recognition, virtual-reality (VR), augmented-reality (AR), interactive gaming, etc. These applications are very often running on mobile devices (MDs); however, the limited computation and processing capability of MDs may degrade the performance measures of the applications and result in the undesired quality of experience (QoE) [1]. Multi-access edge computing (MEC) has emerged as a potent tool to satisfy the growing demands of high task execution rate, low latency, and low energy consumption by bringing the computation and storage resources to the edge of wireless networks, such as the base

stations (BSs) of cellular networks. Leveraging the advanced processing capability of MEC servers, conducting computation offloading which offloads computationally intensive tasks from MDs to MEC servers, and executing user tasks at the servers is highly desired [2, 3].

The problem of computation offloading has been considered extensively in recent research work [4–9]. The authors in [4] formulate the cooperative computation offloading problem (which maximizes the expected long-term reward in terms of service delay) as a Markov decision process (MDP) and propose two intelligent computation offloading algorithms based on soft actor critic (SAC), i.e., centralized SAC offloading and decentralized SAC offloading to solve the problem. A device-centric and risk-based distributed approach is proposed in [5], where the authors exploit game theory to obtain the optimal amount of computation offloading volume. The authors in [6] jointly optimize software caching and computation offloading to minimize the weighted sum energy consumption in a multi-user cache-assisted MEC system and propose an alternating direction method of multipliers (ADMM) and penalty convex–concave procedure (Penalty-CCP) to obtain the sub-optimal solutions. In [7], the authors formulate an energy-efficient computation offloading problem as a mixed integer nonlinear programming problem in a MEC-enabled small cell network. To minimize the energy consumption of all user equipments, a sub-optimal algorithm consisting of a hierarchical genetic algorithm and a particle swarm optimization (PSO)-based computation algorithm is proposed. The authors in [8] study the computation offloading and caching problem aiming at minimizing the execution latency of user tasks by utilizing a collaborative call graph approach. In [9], a collaborative computation offloading scheme is proposed for centralized computing environment and a game-theoretic approach is proposed for distributed computing environment so as to minimize the energy consumption of the system.

To alleviate the ever-growing resource contention and improve the communication and computational efficiency of stand-alone MEC servers, device-to-device (D2D) communication is leveraged, where the diversity among nearby devices can be exploited to share the computational burden [10, 11]. D2D-aided computation offloading techniques have been considered in recent research work [12–15].

Taking into account the dynamic system status and random task arrival rate, the authors in [12] investigate the energy-efficient task offloading problem in socially-aware D2D-assisted MEC networks and maximize the long-term network utility by jointly optimizing D2D connection selection and task allocation. Aiming at minimizing the computation latency for the tasks in a D2D-enabled heterogeneous network, the authors in [13] formulate a user-assisted multi-task offloading problem under the constraints on latency and energy consumption. A distributed optimization scheme based on ADMM is presented to determine the task offloading strategy. The authors in [14] formulate the computation capacity maximization problem in a multi-user D2D-MEC system as a mixed integer nonlinear programming problem with constraints on both communication and computation resources. The original problem is decomposed into two sub-problems, where the first sub-problem aims to minimize the required edge computation resources for a given D2D pair, and the second sub-problem aims to maximize the computation capacity of the D2D-MEC system. A task offloading framework is proposed in [15], where MDs could share their computation and communication resources among

each other via the assistance of network operators. Lyapunov optimization tool is utilized to produce dynamic task offloading decisions, which minimizes the time-averaged energy consumption.

Some research work jointly considers computation offloading and resource allocation problems [16–22]. In [16], the authors jointly optimize task offloading, cache decision, transmission power and central processing unit (CPU) frequency allocation to minimize the weighted sum cost of the execution delay and energy consumption in a cloud-edge heterogeneous network. The authors in [17] consider a D2D-enabled MEC system and formulate user association, task offloading and resource allocation problem as a latency minimization problem. By solving the optimization problem, a joint optimal strategy is obtained. In order to maximize the long-term utility energy efficiency, the authors in [18] jointly optimize the transmit power of the D2D link, the cellular uplink transmit power and the local CPU speed in a wireless powered D2D-enabled MEC system. Lyapunov optimization method is employed to transform the original problem into a series of deterministic drift-plus-penalty sub-problems in each time slot. The authors in [19] propose a mobility-aware task scheduling approach in a D2D-enabled cooperative MEC framework, in which joint optimization of user mobility, computation capacities and task properties is performed to minimize task offloading latency. The authors in [20] jointly optimize computing resource, transmit power and channel allocation to minimize the weighted sum of delay and energy consumption of all users. Due to multivariate fractional summation nature, the original optimization problem is dissolved into sub-problems, namely, power allocation sub-problem, and channel allocation sub-problem. To solve the power allocation sub-problem, the authors employ PSO algorithm, and for the channel allocation sub-problem, a one-to-one matching algorithm based on swapping operations and Pareto improvement is proposed. The authors in [21] formulate the computation offloading and resource allocation problem in a multi-user MEC system as a weighted delay and energy consumption minimization problem and solve the problem by exploiting branch and bound method. A two-stage heuristic optimization (THO) algorithm is proposed in [22], which minimizes the overall energy consumption of the MDs by jointly designing task offloading decisions, channel selection, power allocation and resource allocation strategy.

Emerged as a key enabler for the fifth generation (5G) of wireless networks, non-orthogonal multiple access (NOMA) allows multiple users to share the same resource block (RB) simultaneously by enabling superposition coding (SC) at the transmitters and successive interference cancellation (SIC) at the receivers [23, 24]. Aiming to achieve high spectral efficiency, improved quality of service (QoS) and lower latency, some recent research work has been carried out considering the cooperation of NOMA and MEC [25–33].

In order to minimize the overall system delay, the authors in [25] jointly optimize the offloaded computation-workloads and the transmission time in a NOMA-assisted MEC system. To solve the formulated joint optimization problem, two algorithms are proposed for single-user case and multi-user case, respectively. The authors in [26] jointly optimize the MEC server allocation, the transmit power allocation of all the MDs, the transmit power allocation of the MEC servers, the computational resource allocation of the MEC servers, the time allocation and the channel allocation variables to minimize

the overall delay of all the tasks. The authors in [27] propose a hybrid NOMA-MEC offloading strategy and formulate a multi-objective optimization problem to minimize the user's energy consumption by finding low-complexity pareto-optimal resource allocation solution using exhaustive search method. Taking into account two different wireless channel scenarios, namely, static-channel scenario and dynamic-channel scenario, the authors in [28] propose an algorithm based on channel quality ranking (CQR) as a means to minimize the overall computation delay for a single-user multi-edge computing server. An optimal offloading solution is obtained by combining the golden section search method with the CQR algorithm for a static channel. For a dynamic channel, an algorithm based on deep reinforcement learning (DRL) is proposed. A hybrid NOMA-MEC offloading framework is proposed in [29], where the authors jointly optimize power, time and sub-channels to minimize the overall energy consumption. A switched hybrid NOMA scheme is proposed to allocate power and time, while the total reward exchange stable algorithm is used for channel allocation.

Exploiting the advantages of NOMA-based MEC communication in vehicular networks, the authors in [30] propose a NOMA-enabled vehicular edge computing (VEC) network, in which the joint optimization of offloading decisions, vehicular user equipment clustering, sub-channel allocation, computational resource allocation and transmit power control is implemented to minimize the overall system cost. A backscatter-assisted wireless-powered NOMA-MEC framework is presented in [31], in which the authors optimize energy harvesting (EH) time, backscatter communication time, uplink time, power reflection coefficient and transmit power, as well as computing frequencies in order to maximize the total amount of computation bits across all internet of things (IoT) devices. The authors in [32] jointly optimize the offloaded computation workloads of users, the offloading-duration as well as the computation resource allocation of the MEC servers to minimize the overall task execution latency. A computation efficiency maximization problem is formulated for NOMA-enabled MEC networks in [33], where the authors jointly optimize the transmit power of users and the CPU frequency of MEC servers. To analyze the impacts of delay and energy consumption on computation offloading and resource allocation, the authors in [34] formulate a joint latency and energy consumption minimization problem and provide analytical results for both NOMA uplink and downlink communication scenarios. Aiming at minimizing the overall system cost, the authors in [35] jointly optimize the computation-resource allocation at the MEC servers, the MDs computation offloading and the radio resource allocation for the data transmission in a NOMA-enabled multi-access MEC network and propose a three-layered algorithm to obtain the optimal solution. To minimize the weighted sum energy consumption of all the MDs, the authors in [36] jointly optimize task offloading, channel allocation and time allocation.

While computation offloading problem in NOMA-enabled MEC systems has been investigated, the extensive study on the computation offloading and access mode selection problem is missing. In particular, exploiting the advantages of both NOMA and orthogonal multiple access (OMA) technologies so as to enhance the task transmission performance is still an important issue worthy studying. Furthermore,

although there exist numerous research activities that investigate D2D-based computation offloading and NOMA-enabled MEC system, previous studies fail to jointly consider the computation offloading mode, resource allocation and access scheme selection issues for OMA/NOMA-enabled cellular D2D systems. In this paper, we study computation offloading problem in cellular D2D systems. Specifically, it is assumed that MDs may execute their tasks locally, offload their tasks to the MEC servers or to their D2D peers by applying either OMA or NOMA schemes, and the resources of BSs and MEC servers are shared among multiple MDs, we address the computation offloading mode, communication and computation resource sharing and access scheme selection problem. The overall energy consumption is examined and the joint optimization problem is formulated and solved by dividing the original problem into two sub-problems and solving the two sub-problems, respectively. In a nutshell, the key contributions of the proposed design can be summarized as follows:

- To unlock the true potential of key multiple access techniques, i.e., OMA and NOMA, in the perspective to enhance the task transmission performance, which aimed at minimizing the overall energy consumption, in this paper, we consider a D2D-assisted and NOMA-empowered MEC framework, in which computation and communication resources are jointly optimized.
- To preserve the computation and communication efficiency, we jointly investigate the computation offloading mode, resource allocation and access scheme selection issues for OMA/NOMA-empowered cellular D2D system. Each MD in the network can execute its task in one of the three execution modes, i.e., local, MEC or D2D. For the MDs selecting D2D offloading mode, we further invoke OMA and NOMA modes, which allow the MDs to offload their tasks to their D2D peers by applying either OMA or NOMA.
- The joint computation offloading mode selection, resource allocation and access scheme selection problem is formulated as an energy consumption minimization problem. Since the formulated problem is of NP-hard nature which is very difficult to solve in polynomial time, we start from a simple case and consider only OMA-based transmission scheme. We transform the original problem into two sub-problems, i.e., computation offloading sub-problem and resource allocation sub-problem and propose two heuristics to solve them, then for the MDs selecting D2D offloading mode, we conduct user pairing and apply NOMA scheme. Extensive numerical results are provided to validate the performance of the proposed scheme.

The rest of this paper is organized as follows. Section 2 describes the system model. Section 3 further explores the system model, express different delay and energy consumption formulations for various computing modes and formulate the optimization problem. Sections 4 and 5 present the solution of the optimization problem and propose two heuristics to solve it. Extensive simulation results are presented in Sect. 6. A brief discussion on computational complexity and convergence analysis is provided in Sect. 7. Finally, conclusion is drawn and presented in Sect. 8.

2 System model

In this section, we discuss the system model considered in this paper, including network model and communication resources sharing schemes. Table1 summarizes the notations used in this work.

2.1 Network model

We consider a cellular D2D communication system which consists of N BSs and a number of mobile devices (MDs), where each BS is equipped with one MEC server which is capable of offering computation offloading service to the MDs. The MDs in the network can be classified into two types, i.e., task offloading request users (RUs) and service providing users (PUs). Each RU has a computation-intensive task to execute, while each PU is of relatively advanced computation performance and may offer computation offloading service to RUs via D2D links. Denote BS_n as the n-th BS, $1 \leq n \leq N$. For convenience, we denote the MEC server attached to BS_n as MEC_n. Let M and J denote, respectively, the number of RUs and PUs, and RU_m and PU_j denote, respectively, the m-th RU and the j-th PU, $1 \leq m \leq M, 1 \leq j \leq J$.

Let T_m denote the task of RU_m, $1 \leq m \leq M$. We assume that T_m can be described by a triple $< \xi_m, \eta_m, D_m^{\max} >$, where ξ_m is the input data size of T_m, η_m is the computing capacity (in CPU cycles per bit) required to process T_m and $D_m^{\max}$ is the maximum tolerable latency to execute T_m. It is apparent that in the considered cellular D2D system, one RU may execute its task in various manners, i.e., local computing, MEC offloading or D2D offloading. Specifically, in local computing mode, the RU executes its entire task locally. In MEC offloading mode, the RU offloads its task to one MEC server for task execution. In D2D offloading mode, the RU offloads its task to one neighboring PU for task execution. The considered system model is shown in Fig. 1.

Table 1 Summary of notations

Notation	Definition
M	Number of RUs
N	Number of BSs
J	Number of PUs
T_m	The task of RU_m
ξ_m	Data size of T_m
η_m	CPU cycles required to process T_m
$D_m^{\max}$	Maximum tolerable latency to execute T_m
K	Total number of sub-channels
W_0	Bandwidth of each sub-channel
$W_n^{\max}$	Number of RUs accessing the BS_n
$R_{m,j}^{\mathrm{d}}$	The achievable data rate of the link between the RU_m and the PU_j
$h_{m,j,k}^{\mathrm{d}}$	The channel gain of the link between the RU_m and the PU_j at the k-th sub-channel
Dist	Distance of D2D transmission
P_m	Transmission power of RU_m
τ_m	Maximum tolerable delay
σ^2	Noise power
f_m^0	Computational capability of RU_m
f_n^{m}	MEC server computational capability
f_j^{d}	Computational capability of PU_j

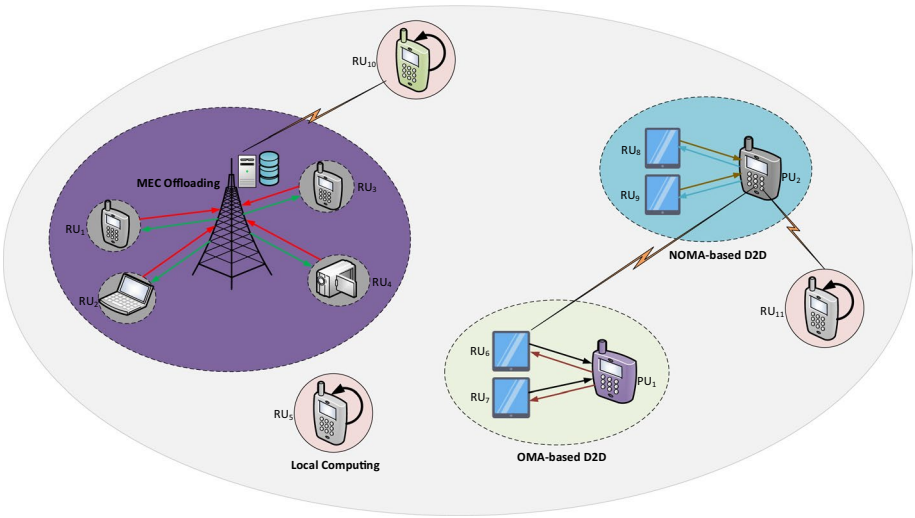

Fig. 1 System model

2.2 Communication resources sharing schemes

To enable efficient task transmission in MEC offloading and D2D offloading mode, we assume that a number of orthogonal sub-channels have been allocated to cellular links and D2D links as well. For cellular link transmission, multiple RUs may access one BS using orthogonal sub-channels and one RU can only occupy one sub-channel for task transmission. Let $W_n^{\max}$ denote the maximal number of sub-channels that can be utilized for data transmission between RUs and BS_n, $1 \leq n \leq N$, and W_0 denote the bandwidth of each sub-channel. Let K denote the total number of sub-channels. Note that in this work, we make a relatively simple assumption on channel interference. That is, we assume that there's no interference between RUs. In the case that there exists interference between different links, we can apply power control or time-frequency resource allocation schemes to migrate or reduce the interference.

For D2D offloading mode, we assume that one PU may assign at most two (adjacent) sub-channels to neighboring RUs in order to enable their task transmission in D2D offloading mode. In the case that two RUs tend to offload their tasks to one PU, given the bandwidth resources of D2D links, the two RUs may access the PU by applying either OMA or NOMA scheme. For convenience, the corresponding task computation modes are referred to as OMA-based D2D offloading mode and NOMA-based D2D offloading mode, respectively. The data rate of the D2D links in both modes is analyzed below.

2.2.1 Data rate in OMA-based D2D offloading mode

To apply OMA scheme to RUs in D2D offloading mode, we assume that one sub-channel is assigned to at most one RU. Suppose RU_m offloads its task T_m to PU_j in OMA-based D2D offloading mode. Let $R_{m,j}^{\mathrm{d}}$ denote the achievable data rate of the link between RU_m and PU_j, $R_{m,j}^{\mathrm{d}}$ can be formulated as

$$R_{m,j}^{\mathrm{d}} = \sum_{k=1}^{K} \mu_{m,j,k}^{\mathrm{d}} W_0 \log_2 \left(1 + \frac{P_m |h_{m,j,k}^{\mathrm{d}}|^2}{\sigma^2}\right), \tag{1}$$

where $\mu_{m,j,k}^{\mathrm{d}} \in \{0,1\}$ is the sub-channel assignment variable in D2D offloading mode, i.e., $\mu_{m,j,k}^{\mathrm{d}} = 1$, if the k-th sub-channel is assigned to RU_m when offloading to PU_j, otherwise, $\mu_{m,j,k}^{\mathrm{d}} = 0$, P_m is the transmit power of RU_m, $h_{m,j,k}^{\mathrm{d}}$ is the channel gain of the link between RU_m and PU_j at the k-th sub-channel, σ^2 is the power of channel noise.

2.2.2 Data rate in NOMA-based D2D offloading mode

Applying NOMA-based D2D offloading mode, we assume that two RUs are allowed to offload their tasks to one PU simultaneously. Suppose RU_m and RU_{m_1} both offload their tasks to PU_j using NOMA scheme. Let $R_{m,m_1,j}^1$ and $R_{m,m_1,j}^2$ denote, respectively, the data rate of the link between RU_m and PU_j, and that between RU_{m_1} and PU_j. Let $h_{m,j,k}^{\mathrm{n}}$ and $h_{m_1,j,k}^{\mathrm{n}}$ be the channel gain of the link between RU_m and PU_j, and that between RU_{m_1} and PU_j at the k-th sub-channel. Without loss of generality, we assume that $|h_{m,j,k}^{\mathrm{n}}| < |h_{m_1,j,k}^{\mathrm{n}}|$, $|h_{m,j,k}^{\mathrm{n}}| \approx |h_{m,j,k+1}^{\mathrm{n}}|$, and $|h_{m_1,j,k}^{\mathrm{n}}| \approx |h_{m_1,j,k+1}^{\mathrm{n}}|$ [37].

Suppose SIC scheme is exploited at PU_j, $R_{m,m_1,j}^1$ and $R_{m,m_1,j}^2$ can be computed, respectively, as

$$R_{m,m_1,j}^1 = 2\sum_{k=1}^{K} \mu_{m,m_1,j,k}^{\mathrm{n}} W_0 \log_2 \left(1 + \frac{P_m |h_{m,j,k}^{\mathrm{n}}|^2}{P_{m_1} |h_{m_1,j,k}^{\mathrm{n}}|^2 + \sigma^2}\right), \tag{2}$$

$$R_{m,m_1,j}^2 = 2\sum_{k=1}^{K} \mu_{m,m_1,j,k}^{\mathrm{n}} W_0 \log_2 \left(1 + \frac{P_{m_1} |h_{m_1,j,k}^{\mathrm{n}}|^2}{\sigma^2}\right), \tag{3}$$

where $\mu_{m,m_1,j,k}^{\mathrm{n}} \in \{0,1\}$ is the sub-channel assignment variable in NOMA-based D2D offloading mode, i.e, if the k-th and the $(k+1)$-th sub-channel are allocated to RU_m and RU_{m_1} for transmitting tasks to PU_j in NOMA-based D2D offloading mode, we set $\mu_{m,m_1,j,k}^{\mathrm{n}} = 1$, otherwise, $\mu_{m,m_1,j,k}^{\mathrm{n}} = 0$.

2.3 Task execution cost in various computation modes

In this section, the delay and energy consumption for task execution in different computation modes are analyzed.

2.3.1 Local computing mode

In the case that RU_m executes its task locally, $1 \le m \le M$, the task execution delay can be characterized by

$$D_m^0 = \frac{\eta_m}{f_m^0}, \tag{4}$$

where f_m^0 denotes the computational capability of RU_m. The energy consumption of RU_m due to task execution can be expressed as

$$E_m^0 = \kappa_m \eta_m (f_m^0)^2, \tag{5}$$

where κ_m is the energy consumption coefficient of RU_m, which depends on the attributes of the CPU of RU_m [21].

2.3.2 MEC offloading mode

In MEC offloading mode, one RU sends its task to one of the MEC servers, which then conducts task execution for the RU. Hence, the delay required to complete task execution can be computed as the sum of task transmission time from the RU to the MEC server, and the task execution time at the MEC server. Suppose RU_m offloads its task T_m to MEC_n, the total time for completing task execution can be calculated as

$$D_{m,n}^{\mathrm{m}} = D_{m,n}^{\mathrm{m,t}} + D_{m,n}^{\mathrm{m,e}}, \tag{6}$$

where $D_{m,n}^{\mathrm{m,t}}$ and $D_{m,n}^{\mathrm{m,e}}$ denote, respectively, the transmission time and execution time of T_m. It should be noted that after executing T_m, MEC_n needs to transmit the result back to RU_m. Since the data size of the task after execution is in general very small, the required transmission delay from MEC_n to RU_m is negligible [21].

$D_{m,n}^{\mathrm{m,t}}$ in (6) can be formulated as

$$D_{m,n}^{\mathrm{m,t}} = \frac{\xi_m}{R_{m,n}^{\mathrm{m}}}, \tag{7}$$

where $R_{m,n}^{\mathrm{m}}$ is the transmission rate of the link between RU_m and MEC_n, which can be expressed as

$$R_{m,n}^{\mathrm{m}} = \sum_{k=1}^{K} \mu_{m,n,k}^{\mathrm{m}} W_0 \log_2 \left(1 + \frac{P_m |h_{m,n,k}^{\mathrm{m}}|^2}{\sigma^2}\right), \tag{8}$$

where $\mu_{m,n,k}^{\mathrm{m}} \in \{0, 1\}$ is the sub-channel allocation variable in MEC offloading mode, i.e., $\mu_{m,n,k}^{\mathrm{m}} = 1$, if the k-th sub-channel is allocated to RU_m for offloading its task to MEC_n, otherwise, $\mu_{m,n,k}^{\mathrm{m}} = 0$, $h_{m,n,k}^{\mathrm{m}}$ denotes the channel gain of the link between RU_m and MEC_n at the k-th sub-channel.

The task execution delay, denoted by $D_{m,n}^{\mathrm{m,e}}$ in (6), can be characterized as

$$D_{m,n}^{\mathrm{m,e}} = \frac{\eta_m}{f_n^{\mathrm{m}}}, \tag{9}$$

where f_n^{m} denotes the computational capacity of MEC_n for processing the task of one RU.

The energy consumption in MEC offloading mode is resulted from task transmission and execution. Consider RU_m offloads its task to MEC_n, we obtain the energy consumption as

$$E_{m,n}^{\mathrm{m}} = E_{m,n}^{\mathrm{m,t}} + E_{m,n}^{\mathrm{m,e}}, \tag{10}$$

where $E_{m,n}^{\mathrm{m,t}}$ is the energy consumption of RU_m when transmitting its task to MEC_n, which is given by

$$E_{m,n}^{\mathrm{m,t}} = P_m D_{m,n}^{\mathrm{m,t}}. \tag{11}$$

$E_{m,n}^{\mathrm{m,e}}$ in (10) is the energy consumption of MEC_n when executing T_m for RU_m. $E_{m,n}^{\mathrm{m,e}}$ can be computed as

$$E_{m,n}^{\mathrm{m,e}} = \kappa_n^{\mathrm{m}} \eta_m (f_n^{\mathrm{m}})^2, \tag{12}$$

where κ_n^{m} denotes the energy consumption coefficient of MEC_n.

2.3.3 D2D offloading mode

In D2D offloading mode, one RU may transmit its task to a neighboring PU which then executes the tasks for the RU. In order to facilitate efficient spectrum utilization, we assume that both OMA-based D2D scheme and NOMA-based D2D scheme are allowed during the task transmission from the RUs to the PUs.

To apply OMA-based D2D offloading mode, we assume that one sub-channel is assigned to at most one RU for offloading its task to one PU. Suppose RU_m offloads its task T_m to PU_j, the corresponding task completion delay can be determined by

$$D_{m,j}^{\mathrm{d}} = D_{m,j}^{\mathrm{d,t}} + D_{m,j}^{\mathrm{d,e}}, \tag{13}$$

where $D_{m,j}^{\mathrm{d,t}}$ and $D_{m,j}^{\mathrm{d,e}}$ denote, respectively, the transmission time required when RU_m offloads its task T_m to PU_j and the execution time of task T_m at PU_j.

$D_{m,j}^{\mathrm{d,t}}$ can be expressed as

$$D_{m,j}^{\mathrm{d,t}} = \frac{\xi_m}{R_{m,j}^{\mathrm{d}}}. \tag{14}$$

$D_{m,j}^{\mathrm{d,e}}$ in (13) can be characterized by

$$D_{m,j}^{\mathrm{d,e}} = \frac{\eta_m}{f_j^{\mathrm{d}}}, \tag{15}$$

where f_j^{d} is the computational capacity of PU_j for processing the task of one RU. The energy consumption in D2D offloading mode is caused by task transmission and execution. When RU_m offloads its task T_m to PU_j, the energy consumption is given by

$$E_{m,j}^{\mathrm{d}} = E_{m,j}^{\mathrm{d,t}} + E_{m,j}^{\mathrm{d,e}}, \tag{16}$$

where $E_{m,j}^{\mathrm{d,t}}$ and $E_{m,j}^{\mathrm{d,e}}$ denote the energy consumption of RU_m for task transmission and the energy consumption of PU_j for task execution, respectively. $E_{m,j}^{\mathrm{d,t}}$ can be expressed as

$$E_{m,j}^{\mathrm{d,t}} = P_m D_{m,j}^{\mathrm{d,t}}, \tag{17}$$

$E_{m,j}^{\mathrm{d,e}}$ is given by

$$E_{m,j}^{\mathrm{d,e}} = \kappa_j^{\mathrm{d}} \eta_m (f_j^{\mathrm{d}})^2, \tag{18}$$

where κ_j^{d} denotes the energy consumption coefficient of PU_j.

To apply NOMA-based D2D offloading mode, we assume that two RUs offload their tasks to one PU simultaneously. Suppose RU_m and RU_{m_1} both offload their tasks to PU_j using two adjacent sub-channels, the task completion time can be expressed as

$$D^{\mathrm{n}}_{m,m_1,j} = D^{\mathrm{n,t}}_{m,m_1,j} + D^{\mathrm{n,e}}_{m,m_1,j}, \tag{19}$$

where $D^{\mathrm{n,t}}_{m,m_1,j}$ and $D^{\mathrm{n,e}}_{m,m_1,j}$ are, respectively, the task transmission time and execution time of RU_m and RU_{m_1} when offloading their tasks to PU_j. $D^{\mathrm{n,t}}_{m,m_1,j}$ is given by

$$D^{\mathrm{n,t}}_{m,m_1,j} = D^{\mathrm{n,t,1}}_{m,m_1,j} + D^{\mathrm{n,t,2}}_{m,m_1,j}, \tag{20}$$

where $D^{\mathrm{n,t,1}}_{m,m_1,j}$ and $D^{\mathrm{n,t,2}}_{m,m_1,j}$ denote, respectively, the task transmission time of RU_m and RU_{m_1}, and can be computed as

$$D^{\mathrm{n,t,1}}_{m,m_1,j} = \frac{\xi_m}{R^1_{m,m_1,j}}, \tag{21}$$

$$D^{\mathrm{n,t,2}}_{m,m_1,j} = \frac{\xi_{m_1}}{R^2_{m,m_1,j}}. \tag{22}$$

The task execution time of RU_m and RU_{m_1} at PU_j denoted by $D^{\mathrm{n,e}}_{m,m_1,j}$ in (19) can be calculated as

$$D^{\mathrm{n,e}}_{m,m_1,j} = D^{\mathrm{n,e,1}}_{m,m_1,j} + D^{\mathrm{n,e,2}}_{m,m_1,j}, \tag{23}$$

where $D^{\mathrm{n,e,1}}_{m,m_1,j}$ and $D^{\mathrm{n,e,2}}_{m,m_1,j}$ are, respectively, the task execution time of RU_m and RU_{m_1} at PU_j, which are given by

$$D^{\mathrm{n,e,1}}_{m,m_1,j} = \frac{\eta_m}{f^{\mathrm{d}}_j}, \tag{24}$$

$$D^{\mathrm{n,e,2}}_{m,m_1,j} = \frac{\eta_{m_1}}{f^{\mathrm{d}}_j}. \tag{25}$$

The energy consumed due to task transmission and execution when RU_m and RU_{m_1} offloading their tasks to PU_j in NOMA-based D2D mode can be expressed as

$$E^{\mathrm{n}}_{m,m_1,j} = E^{\mathrm{n,t}}_{m,m_1,j} + E^{\mathrm{n,e}}_{m,m_1,j}, \tag{26}$$

where $E^{\mathrm{n,t}}_{m,m_1,j}$ and $E^{\mathrm{n,e}}_{m,m_1,j}$ denote, respectively, the energy consumption during transmission and that during task execution. $E^{\mathrm{n,t}}_{m,m_1,j}$ is given by

$$E^{\mathrm{n,t}}_{m,m_1,j} = E^{\mathrm{n,t,1}}_{m,m_1,j} + E^{\mathrm{n,t,2}}_{m,m_1,j}, \tag{27}$$

where $E^{\mathrm{n,t,1}}_{m,m_1,j}$ and $E^{\mathrm{n,t,2}}_{m,m_1,j}$ are, respectively, the transmission energy consumption of RU_m and RU_{m_1}, and can be expressed as

$$E^{\mathrm{n,t,1}}_{m,m_1,j} = P_m D^{\mathrm{n,t,1}}_{m,m_1,j} \tag{28}$$

$$E^{\mathrm{n,t},2}_{m,m_1,j} = P_{m_1} D^{\mathrm{n,t},2}_{m,m_1,j}. \tag{29}$$

$E^{\mathrm{n,e}}_{m,m_1,j}$ can be expressed as

$$E^{\mathrm{n,e}}_{m,m_1,j} = E^{\mathrm{n,e},1}_{m,m_1,j} + E^{\mathrm{n,e},2}_{m,m_1,j}, \tag{30}$$

where $E^{\mathrm{n,e},1}_{m,m_1,j}$ and $E^{\mathrm{n,e},2}_{m,m_1,j}$ are, respectively, the energy consumption of PU_j when executing the task of RU_m and RU_{m_1}. $E^{\mathrm{n,e},1}_{m,m_1,j}$ and $E^{\mathrm{n,e},2}_{m,m_1,j}$ are given by

$$E^{\mathrm{n,e},1}_{m,m_1,j} = \kappa^{\mathrm{d}}_j \eta_m (f^{\mathrm{d}}_j)^2, \tag{31}$$

$$E^{\mathrm{n,e},2}_{m,m_1,j} = \kappa^{\mathrm{d}}_j \eta_{m_1} (f^{\mathrm{d}}_j)^2. \tag{32}$$

3 Delay and energy consumption function formulation

In this work, we design a centralized scheme, where instead of optimizing the performance of a particular RU, we aim to minimize the overall system energy consumption which consists of the local task execution energy consumption of the RUs, the task transmission energy consumption of the RUs and the task execution energy consumption of the PUs and the MEC servers. In this section, we first examine the total energy consumption, then formulate joint computation offloading and resource allocation strategy as an energy consumption minimization problem.

The total energy consumption required for transmitting and executing the tasks of all the RUs can be expressed as

$$E = \sum_{m=1}^{M} E_m, \tag{33}$$

where E_m is the energy consumption required for transmitting and executing the task of RU_m, and can be computed as

$$\begin{aligned} E_m = {} & x^0_m E^0_m + \sum_{n=1}^{N} x^{\mathrm{m}}_{m,n} E^{\mathrm{m}}_{m,n} + \sum_{j=1}^{J} x^{\mathrm{d}}_{m,j} E^{\mathrm{d}}_{m,j} \\ & + \sum_{m_1=1, m_1 \neq m}^{M} \sum_{j=1}^{J} x^{\mathrm{n}}_{m,m_1,j} E^{\mathrm{n}}_{m,m_1,j}, \end{aligned} \tag{34}$$

where $x^0_m \in \{0,1\}$ denotes the computation offloading decision variable for local computing mode, i.e., x^0_m=1, if RU_m executes its task locally, otherwise, $x^0_m = 0$; $x^{\mathrm{m}}_{m,n} \in \{0,1\}$ is the computation offloading decision variable for MEC offloading mode, if RU_m offloads its task to MEC_n, $x^{\mathrm{m}}_{m,n} = 1$, otherwise, $x^{\mathrm{m}}_{m,n} = 0$. Likewise, if RU_m offloads its task to PU_j in OMA-based D2D mode, $x^{\mathrm{d}}_{m,j} = 1$, otherwise, $x^{\mathrm{d}}_{m,j} = 0$, whereas $x^{\mathrm{n}}_{m,m_1,j} = 1$ indicates that both RU_m and RU_{m_1} offload their tasks to PU_j using NOMA-based D2D scheme, otherwise, $x^{\mathrm{n}}_{m,m_1,j} = 0$.

3.1 Optimization constraints

In order to jointly design the computation offloading and resource allocation strategy, we consider a number of optimization constraints.

3.1.1 Delay constraint

The tasks of RUs should be executed before the given maximum deadline, i.e.,

$$\mathrm{C1}: D_m \leq D_m^{\max}, \quad \forall m. \tag{35}$$

where D_m is the time required for transmitting and executing the task of RU_m, and can be computed as

$$\begin{aligned} D_m = x_m^0 D_m^0 + \sum_{n=1}^{N} x_{m,n}^{\mathrm{m}} D_{m,n}^{\mathrm{m}} + \sum_{j=1}^{J} x_{m,j}^{\mathrm{d}} D_{m,j}^{\mathrm{d}} \\ + \sum_{m_1=1, m_1 \neq m}^{M} \sum_{j=1}^{J} x_{m,m_1,j}^{\mathrm{n}} D_{m,m_1,j}^{\mathrm{n}}. \end{aligned} \tag{36}$$

3.1.2 Computation offloading constraint

We assume that each RU can only execute its task in one of the three offloading modes, i.e., local computing, MEC offloading or D2D offloading, hence, the computing mode selection constraint is given as

$$\mathrm{C2}: x_m^0 + \sum_{n=1}^{N} x_{m,n}^{\mathrm{m}} + \sum_{j=1}^{J} x_{m,j}^{\mathrm{d}} + \sum_{m_1=1, m_1 \neq m}^{M} \sum_{j=1}^{J} x_{m,m_1,j}^{\mathrm{n}} \leq 1. \tag{37}$$

3.1.3 Resource allocation constraints in MEC offloading mode

In MEC offloading mode, we assume that one sub-channel can only be assigned to one RU and vice versa, hence, the sub-channel allocation constraints can be expressed as

$$\mathrm{C3}: \sum_{m=1}^{M} \sum_{n=1}^{N} \mu_{m,n,k}^{\mathrm{m}} \leq 1, \quad \forall k, \tag{38}$$

$$\mathrm{C4}: \sum_{k=1}^{K} \mu_{m,n,k}^{\mathrm{m}} \leq 1, \quad \forall m, n. \tag{39}$$

The maximal number of sub-channels of BS_n puts the constraint on the number of RUs accessing the BS, i.e.,

$$\mathrm{C5}: \sum_{m=1}^{M} \sum_{k=1}^{K} \mu_{m,n,k}^{\mathrm{m}} \leq W_n^{\max}, \quad \forall n. \tag{40}$$

3.1.4 Resource allocation constraints in OMA-based D2D scheme

In OMA-based D2D scheme, we assume that one sub-channel can only be assigned to one RU and vice versa, hence, the sub-channel allocation constraints can be expressed as

$$\mathrm{C6}: \sum_{m=1}^{M} \sum_{j=1}^{J} \mu_{m,j,k}^{\mathrm{d}} \leq 1, \quad \forall k, \tag{41}$$

$$\mathrm{C7}: \sum_{k=1}^{K} \mu_{m,j,k}^{\mathrm{d}} \leq 1, \quad \forall m, j. \tag{42}$$

In OMA-based D2D scheme, at most two RUs may access one PU for computation offloading utilizing two sub-channels, we obtain the following constraint:

$$\mathrm{C8}: \sum_{m=1}^{M} \sum_{k=1}^{K} \mu_{m,j,k}^{\mathrm{d}} \leq 2, \quad \forall j. \tag{43}$$

3.1.5 Resource allocation constraints in NOMA-based D2D scheme

In NOMA-based D2D scheme, each sub-channel can only be assigned to one NOMA pair, we obtain

$$\mathrm{C9}: \sum_{m=1}^{M} \sum_{m_1=1, m_1 \neq m}^{M} \sum_{j=1}^{J} \mu_{m,m_1,j,k}^{\mathrm{n}} \leq 1, \quad \forall k. \tag{44}$$

In NOMA-based D2D scheme, at most two sub-channels are assigned to two RUs, i.e.,

$$\mathrm{C10}: \sum_{m=1}^{M} \sum_{m_1=1, m_1 \neq m}^{M} \sum_{k=1}^{K} \mu_{m,m_1,j,k}^{\mathrm{n}} \leq 2, \quad \forall j. \tag{45}$$

We assume that two adjacent sub-channels should be assigned to one NOMA pair, i.e.,

$$\mathrm{C11}: \mu_{m,m_1,j,k}^{\mathrm{n}} \odot \mu_{m,m_1,j,k+1}^{\mathrm{n}} = 1, \quad \forall m, m_1, j, k < K, \tag{46}$$

where $\odot$ represents the inclusive OR operator.

3.1.6 Constraints on offloading mode selection and resource allocation

Apparently, there exists a direct relation between offloading mode selection and sub-channel allocation decision in all the three offloading modes, we express the constraints as follows:

$$\mathrm{C12}: x_{m,n}^{\mathrm{m}} \odot \sum_{k=1}^{K} \mu_{m,n,k}^{\mathrm{m}} = 1, \quad \forall m, n, \tag{47}$$

$$\mathrm{C13}: x_{m,j}^{\mathrm{d}} \odot \sum_{k=1}^{K} \mu_{m,j,k}^{\mathrm{d}} = 1, \quad \forall m, j, \tag{48}$$

$$\mathrm{C14}: x^{\mathrm{n}}_{m,m_1,j} \odot \sum_{k=1}^{K} \mu^{\mathrm{n}}_{m,m_1,j,k} = 1, \quad \forall m, m_1, j. \tag{49}$$

3.2 Optimization problem formulation

To minimize the energy consumption subject to a number of constraints, we formulate the optimization problem as follows:

$$\begin{aligned} &\min_{x^0_m, x^{\mathrm{m}}_{m,n}, x^{\mathrm{d}}_{m,j}, x^{\mathrm{n}}_{m,m_1,j}, \mu^{\mathrm{m}}_{m,n,k}, \mu^{\mathrm{d}}_{m,j,k}, \mu^{\mathrm{n}}_{m,m_1,j,k}} \quad E \\ &\text{s.t. } \mathrm{C1} - \mathrm{C14}. \end{aligned} \tag{50}$$

4 Proposed algorithm: no NOMA scheme applied

Since the optimization problem formulated in (50) is NP hard, which is inconvenient to solve in polynomial time. In this section, we start from a relatively simple case, i.e., for D2D offloading mode, only OMA-based transmission scheme is considered, and propose a heuristic algorithm. By examining the energy consumption of RUs in different task offloading modes, we first determine local computing mode, then present a priority-based sub-channel allocation algorithm for conflicting RUs. In next section, we consider the RUs choosing D2D offloading mode, and determine task offloading mode and sub-channel allocation strategy.

4.1 Rewriting energy consumption in various offloading modes

To minimize the energy consumption in (50), we may examine extensively the energy consumption of individual RUs in different offloading modes at various sub-channels. Let E^{loc}_m, $E^{\mathrm{mec}}_{m,n,k}$, $E^{\mathrm{d2d}}_{m,j,k}$ denote, respectively, the energy consumption of RU_m in local computing mode, MEC offloading mode and OMA-based D2D offloading mode.

Suppose that only OMA scheme is allowed in D2D offloading mode and taking into account the constraints on mode selection variables and sub-channel allocation variables specified in C12, C13, we may rewrite the energy consumption E as follows:

$$\begin{aligned} E = &\sum_{m=1}^{M} x^0_m E^{\mathrm{loc}}_m + \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{k=1}^{K} \mu^{\mathrm{m}}_{m,n,k} E^{\mathrm{mec}}_{m,n,k} \\ &+ \sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{K} \mu^{\mathrm{d}}_{m,j,k} E^{\mathrm{d2d}}_{m,j,k}. \end{aligned} \tag{51}$$

The original optimization problem in (50) is reduced to

$$\begin{aligned} &\min_{x^0_m, \mu^{\mathrm{m}}_{m,n,k}, \mu^{\mathrm{d}}_{m,j,k}} \quad E \\ &\text{s.t. } \mathrm{C2}': x^0_m + \sum_{n=1}^{N}\sum_{k=1}^{K} \mu^{\mathrm{m}}_{m,n,k} + \sum_{j=1}^{J}\sum_{k=1}^{K} \mu^{\mathrm{d}}_{m,j,k} \le 1, \\ &\mathrm{C3}-\mathrm{C8}. \end{aligned} \tag{52}$$

The above optimization problem involves computation mode selection and sub-channel allocation among various offloading modes, which is still difficult to tackle. In this subsection, we propose a heuristic algorithm, which conducts the following steps successively, i.e., local computing mode selection, sub-channel allocation for non-conflicting RUs, priority-based sub-channel allocation for conflicting RUs.

4.2 Local computing mode selection

For RU_m, we may calculate its energy consumption in different computing modes at different sub-channels. It is obvious that if one RU needs to consume the minimum energy when performing local computing compared with both the MEC offloading mode and the D2D offloading mode, the RU should execute its task locally. Therefore, we can first assign the local computing mode for the RUs by comparing its energy consumption in various computing modes. That is, if RU_m achieves the minimum energy consumption when executing its task locally, i.e., $E_m^{\mathrm{loc}} \leq E_{m,n,k}^{\mathrm{mec}}$ and $E_m^{\mathrm{loc}} \leq E_{m,j,k}^{\mathrm{d2d}}$, $\forall n, j, k$, we should assign the local computing mode to RU_m, i.e., $x_m^{0,*} = 1$, $x_{m,n}^{\mathrm{m},*} = 0$, $x_{m,j}^{\mathrm{d},*} = 0$, where $x_m^{0,*}$, $x_{m,n}^{\mathrm{m},*}$, and $x_{m,j}^{\mathrm{d},*}$ represent the optimal computing and offloading strategy.

4.3 K-M algorithm-based sub-channel allocation for nonconflicting RUs

After removing the RUs which have been assigned local computing mode, we place the remaining RUs into a set, denoted by Ψ_{RU}. We now solve the optimization problem in (52) for the RUs in Ψ_{RU}.

It is noticeable that the formulated optimization problem is similar as a matching problem in a bipartite graph, however, it is not a typical one-to-one matching problem as the sub-channel allocation among different offloading modes should be taken into account. To tackle this problem, we first consider an ideal sub-channel allocation assumption for both the BSs and the PUs. More specifically, we assume that all the sub-channels are available for all the BSs and the PUs, and then determine the resource allocation and computation offloading strategy which minimizes the energy consumption. Equivalently, we virtualize the set of sub-channels into $N + J$ sets and assign each BS and PU one set of sub-channels. For instance, BS_n is assigned the $((n-1)K+1)$-th to the (nK)-th sub-channels, and PU_j is assigned $((N+j-1)K+1)$-th to the $((N+j)K)$-th sub-channels, $1 \leq n \leq N, 1 \leq k \leq K$.

The energy consumption E can then be rewritten as

$$\begin{aligned}\bar{E} = &\sum_{m=1}^{M} x_m^0 E_m^{\mathrm{loc}} + \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{k'=(n-1)K+1}^{nK} \bar{\mu}_{m,n,k'}^{\mathrm{m}} \bar{E}_{m,n,k'}^{\mathrm{mec}} \\ &+ \sum_{m=1}^{M}\sum_{j=1}^{J}\sum_{k'=(N+j-1)K+1}^{(N+j)K} \bar{\mu}_{m,j,k'}^{\mathrm{d}} \bar{E}_{m,j,k'}^{\mathrm{d2d}},\end{aligned} \tag{53}$$

where $\bar{E}_{m,n,k'}^{\mathrm{mec}}$ is the energy consumption of RU_m when offloading its task to MEC_n using the k'-th sub-channel after sub-channel virtualization, $\bar{E}_{m,n,k'}^{\mathrm{mec}}$ can be expressed as

$$\bar{E}_{m,n,k'}^{\mathrm{mec}} = E_{m,n,k}^{\mathrm{mec}}, 1 \leq k \leq K, k' = (n-1)K + k. \tag{54}$$

$\bar{E}_{m,j,k'}^{\mathrm{d2d}}$ is the energy consumption of RU_m when offloading its task to PU_j using the k'-th sub-channel after sub-channel virtualization, $\bar{E}_{m,j,k'}^{\mathrm{d2d}}$ can be expressed as

$$\bar{E}^{\mathrm{d2d}}_{m,j,k'} = E^{\mathrm{d2d}}_{m,j,k}, 1 \le k \le K, k' = (N+j-1)K + k. \tag{55}$$

Similarly, $\bar{\mu}^{\mathrm{m}}_{m,n,k'} = \mu^{\mathrm{m}}_{m,n,k}$ for $k' = (n-1)K + k$, and $\bar{\mu}^{\mathrm{d}}_{m,j,k'} = \bar{\mu}^{\mathrm{d}}_{m,j,k}$, for $k' = (N+j-1)K + k$.

The original optimization problem in (52) can be expressed as

$$\begin{aligned}
&\min_{x^0_m, \bar{\mu}^{\mathrm{m}}_{m,n,k'}, \bar{\mu}^{\mathrm{d}}_{m,j,k'}} \quad \bar{E} \\
\text{s.t.} \quad &\mathrm{C2}' : x^0_m + \sum_{n=1}^{N} \sum_{k'=(n-1)K+1}^{nK} \bar{\mu}^{\mathrm{m}}_{m,n,k'} + \sum_{j=1}^{J} \sum_{k'=(N+j-1)K+1}^{(N+j)K} \bar{\mu}^{\mathrm{d}}_{m,j,k'} \le 1, \\
&\mathrm{C3}' : \sum_{m=1}^{M} \sum_{n=1}^{N} \bar{\mu}^{\mathrm{m}}_{m,n,k'} \le 1, (n-1)K + 1 \le k \le nK, \\
&\mathrm{C4}' : \sum_{k'=(n-1)K+1}^{nK} \bar{\mu}^{\mathrm{m}}_{m,n,k'} \le 1, \quad \forall m, n, \\
&\mathrm{C5}' : \sum_{m=1}^{M} \sum_{k'=(n-1)K+1}^{nK} \bar{\mu}^{\mathrm{m}}_{m,n,k'} \le W^{\max}_n, \quad \forall n, \\
&\mathrm{C6}' : \sum_{m=1}^{M} \sum_{j=1}^{J} \bar{\mu}^{\mathrm{d}}_{m,j,k'} \le 1, (N+j-1)K + 1 \le k' \le (N+j)K, \\
&\mathrm{C7}' : \sum_{k'=(N+j-1)K+1}^{(N+j)K} \bar{\mu}^{\mathrm{d}}_{m,j,k'} \le 1, \quad \forall m, j.
\end{aligned} \tag{56}$$

The above optimization problem can be regarded as a one-to-one matching problem in a bipartite graph, which can be solved by typical algorithm such as the Kuhn–Munkres (K-M) algorithm [38].

Let $\tilde{x}^0_m$, $\tilde{\mu}^{\mathrm{m}}_{m,n,k'}$ and $\tilde{\mu}^{\mathrm{d}}_{m,j,k'}$ denote, respectively, the local optimal strategy of x^0_m, $\bar{\mu}^{\mathrm{m}}_{m,n,k'}$ and $\bar{\mu}^{\mathrm{d}}_{m,j,k'}$ obtained from the K-M algorithm. Based on the local optimal strategy of the RUs, we may check whether there exist non-conflicting RUs of which the selected sub-channel is not shared with other RUs. For non-conflicting RUs, we assign the local optimal offloading and sub-channel allocation strategy as the global optimal one. As an example, suppose the local optimal strategy of RU_{m_1} is $\tilde{x}^0_{m_1} = 0$, $\tilde{\mu}^{\mathrm{m}}_{m_1,n_1,k'_1} = 1$ and $\tilde{\mu}^{\mathrm{d}}_{m_1,j,k'} = 0$, and no other RUs select the same sub-channel, i.e., $\tilde{\mu}^{\mathrm{m}}_{m,n,k'} = 0$ and $\tilde{\mu}^{\mathrm{d}}_{m,j,k'} = 0$, for $m \ne m_1$, $k'_1 \ne k'$, we set the global optimal offloading and sub-channel allocation strategy of RU_{m_1} as $x^{0,*}_{m_1} = 0$, $x^{\mathrm{m},*}_{m_1,n_1} = 1$, and $x^{\mathrm{d},*}_{m_1,j} = 0$, $\mu^{\mathrm{m},*}_{m_1,n_1,k_1} = 1$, $\mu^{\mathrm{m},*}_{m_1,n,k} = 0$ and $\mu^{\mathrm{d},*}_{m_1,j,k} = 0$, for $n \ne n_1$, $k' \ne k$, $k_1 = \mathrm{mod}(k'_1, K)$, $k = \mathrm{mod}(k', K)$, where $\mathrm{mod}(x, y) = x - y\lfloor x/y \rfloor$.

Once the RUs have been assigned global optimal strategy, they are removed from the remaining user set Ψ_{RU} and their selected sub-channels are removed correspondingly.

4.4 Priority-based sub-channel allocation for conflicting RUs

Note that by applying sub-channel virtualization, the BSs and PUs are allowed to share same sub-channels, the obtained local optimal computation offloading and sub-channel

allocation strategy $\tilde{x}_m^0$, $\tilde{\mu}_{m,n,k}^{\mathrm{m}}$ and $\tilde{\mu}_{m,j,k}^{\mathrm{d}}$ may involve resource conflicting among RUs. More specifically, it is probable that more than one RU chooses to occupy a common sub-channel for task offloading. For instance, if $\tilde{\mu}_{m,n,k} = 1$, $\tilde{\mu}_{m_1,n_1,k_1} = 1$, and $\mathrm{mod}(k, K) = \mathrm{mod}(k_1, K)$, then both RU_m and RU_{m_1} choose the $k\mathrm{mod}K$-th sub-channel for task offloading. We refer RU_m and RU_{m_1} as a pair of conflicting users.

Since multiple sub-channels are not allowed for MEC offloading mode and OMA-based D2D offloading mode, we need to design computation offloading and resource allocation strategy for the conflicting RUs. To this end, we propose a priority-based offloading mode selection and sub-channel allocation scheme. The steps of the proposed scheme can be summarized as follows:

(1) Assign priority to the conflicting RUs

We examine the energy consumption of the conflicting RUs and assign various priorities to these RUs. For each RU, we first evaluate the energy consumption in various offloading modes, and set the lowest one as the energy consumption of the RU. Then, aiming to minimize the energy consumption of all the RUs, we order the non-conflicting RUs according to their energy consumption and assign the highest priority to the RU having the lowest energy consumption.

(2) Assign global optimal strategy to the RU with the highest priority

For the RU with the highest priority, the local optimal strategy will be set as its global optimal strategy. We remove this RU as well as the corresponding sub-channel from the RU set and sub-channel set.

(3) Update local optimal strategy of the remaining RUs

The local optimal strategy of the remaining RUs is updated by applying the K-M algorithm. Check whether conflicting RUs exist, if yes, return to (1), otherwise, set the local optimal strategy of the remaining RUs as the global optimal one, and the algorithm terminates.

5 Greedy method-based task offloading and user pairing algorithm: NOMA scheme applied

In this subsection, we consider the RUs which need to offload their tasks to PUs using D2D offloading mode. Since RUs may apply OMA-based D2D scheme or NOMA-based D2D scheme, the optimal computation offloading selection, sub-channel allocation and NOMA paring strategy is very difficult to obtain. For simplicity, we design a greedy-based computation offloading and NOMA paring algorithm.

5.1 Task offloading strategy: one RU case

For individual PUs, we may assign different RUs for conducting D2D offloading, and accordingly, various computation offloading and NOMA paring strategies can be obtained. For one or two RUs, they may choose one PU and offload their tasks to the PU in OFDMA mode. Alternatively, two RUs may form a NOMA pair and send their tasks to a common PU. For a specific PU and the set of RUs choosing the PU to offload tasks, we may list all potential task offloading combinations, and compute the corresponding energy consumption by exploiting extensive search method.

Based on the local optimal strategy of RUs obtained from the K-M algorithm, we assign task offloading strategy for the RUs choosing D2D offloading mode. In the case that only one RU chooses a PU for task offloading, we assign OMA-based D2D offloading mode to the RU. Suppose RU_m is the only user choosing PU_j to offload its task, i.e., $\tilde{\mu}^{\mathrm{d}}_{m,j,k'} = 1$ and $\tilde{\mu}^{\mathrm{d}}_{m',j,k'_1} = 0$, $\forall m' \neq m, \forall k', k'_1$, we set $\mu^{\mathrm{d},*}_{m,j,k} = 1$ and $\mu^{\mathrm{d},*}_{m',j,k_1} = 0$, where $k = \mathrm{mod}(k', K), k_1 = \mathrm{mod}(k'_1, K)$.

If more than one RU choosing one PU to offload their tasks, we need to select one or two RUs and determine the optimal task offloading strategy. To this end, we propose two schemes, i.e., greedy method-based user pairing and task offloading algorithm and low complexity user pairing and task offloading algorithm.

5.2 Task offloading and user pairing algorithm

We assume that multiple RUs select PU_j for task offloading. Let Φ_{RU} denote the set of RUs, i.e., if $\tilde{\mu}^{\mathrm{d}}_{m,j,k'} = 1$, then $\mathrm{RU}_m \in \Phi_{\mathrm{RU}}$. Since at most two RUs are allowed to offload their tasks to one PU, among all the RUs in Φ_{RU}, we need to choose one or two RUs and assign the task offloading mode and the corresponding sub-channel.

5.2.1 Local optimal strategy in OMA-based D2D offloading mode

First examine the optimal performance obtained by using OMA-based D2D offloading mode. Suppose OMA-based D2D offloading mode is assigned to the RUs in Φ_{RU}, we may examine the task offloading performance of the RUs and select one or two RUs achieving the optimal performance. Specifically, for $\forall \mathrm{RU}_m \in \Phi_{\mathrm{RU}}$, compute $E^{\mathrm{d2d}}_{m,j,k}$, and select two RUs with the corresponding sub-channels obtaining the optimal task offloading performance, i.e., if $E^{\mathrm{d2d}}_{m_1,j,k_1} + E^{\mathrm{d2d}}_{m_2,j,k_2} \leq E^{\mathrm{d2d}}_{m'_1,j,k'_1} + E^{\mathrm{d2d}}_{m'_2,j,k'_2}$, for $(m_1, m_2) \neq (m'_1, m'_2)$, $(k_1, k_2) \neq (k'_1, k'_2)$, then RU_{m_1} and RU_{m_2} are selected as the local optimal users and sub-channels k_1 and k_2 should be allocated to the two users. Let $E^{\mathrm{o},*}_{m_1,m_2,j,k_1,k_2} = E^{\mathrm{d2d}}_{m_1,j,k_1} + E^{\mathrm{d2d}}_{m_2,j,k_2}$ denote the local optimal energy consumption of RUs when offloading tasks to PU_j in OMA-based D2D mode.

5.2.2 Local optimal strategy in NOMA-based D2D offloading mode

We then evaluate the task offloading performance in NOMA-based D2D offloading mode. Following a similar manner as in OMA-based case, we choose two RUs in Φ_{RU} to form NOMA pair and compute the task offloading performance on various sub-channels and select the pair achieving the optimal performance.

Let $E^{\mathrm{n},*}_{m'_1,m'_2,j,k}$ denote the energy consumption of $\mathrm{RU}_{m'_1}$ and $\mathrm{RU}_{m'_2}$ when offloading tasks to PU_j in NOMA-based D2D mode via the k-th and $(k+1)$-th sub-channels, $\forall \mathrm{RU}_{m'_1}, \mathrm{RU}_{m'_2} \in \Phi_{\mathrm{RU}}$. If $\tilde{E}^{\mathrm{n}}_{m'_1,m'_2,j,k} \leq \tilde{E}^{\mathrm{n}}_{\bar{m}_1,\bar{m}_2,j,k'}$, $\forall (m'_1, m'_2) \neq (\bar{m}_1, \bar{m}_2)$, $1 \leq k, k' \leq K$, then $\mathrm{RU}_{m'_1}$ and $\mathrm{RU}_{m'_2}$ are selected as the local optimal users in NOMA-based D2D offloading mode.

5.2.3 Determine task offloading and user pairing strategy

Given the local optimal task offloading performance in OMA-based and NOMA-based task offloading modes, we now compare the performance and choose the one offering the better performance. In particular, if the following condition meets:

$$E^{\mathrm{n},*}_{m'_1,m'_2,j,k} \leq E^{\mathrm{o},*}_{m_1,m_2,j,k_1,k_2} \tag{57}$$

we assign NOMA-based D2D offloading mode to $\mathrm{RU}_{m'_1}$ and $\mathrm{RU}_{m'_2}$ with PU_j being the offloading PU, and the k-th and the $(k+1)$-th sub-channels are allocated to the two RUs for task transmission. Similarly, if

$$E^{\mathrm{n},*}_{m'_1,m'_2,j,k} > E^{\mathrm{o},*}_{m_1,m_2,j,k_1,k_2} \tag{58}$$

we assign OMA-based D2D offloading mode to RU_{m_1} and RU_{m_2} with PU_j being the offloading PU and the k_1-th and the k_2-th sub-channels are allocated to the two RUs for task transmission.

Algorithm 1 Greedy-based computation offloading and NOMA pairing algorithm

```
Compute the energy consumption of RUs at each PU in OFDMA mode and NOMA mode
Arrange each PU according to obtained energy consumption metric of different RUs
for all RUs do
    Find particular PU
    j=1
    if (57) holds then
        RU_m, RU_{m_1} will offload at PU_j using NOMA-based D2D offloading mode
    else if (58) holds then
        RU_m, RU_{m_1} will offload at PU_j using OMA-based D2D offloading mode
    end if
    Remove the RUs and PU from the set of RUs and PUs
    j=j+1
end for
```

6 Simulation results

In this section, numerical results are presented to evaluate the performance of the proposed scheme. We run our simulations on Matlab-based simulator. The considered system model is a cellular D2D communication system with 4 BSs, 6 PUs and 5-30 RUs uniformly distributed around the BSs. The overall simulation region is chosen as 1000 m × 1000 m. All the simulation parameters utilized unless explicitly mentioned are reported in Table 2. Results are obtained by averaging over 2000 random trials.

Figure 2 plots the curve for the energy consumption versus the number of RUs when three different algorithms are applied. For comparison, in addition to exhibiting the performance of our proposed algorithm, we also consider the one only OMA-based D2D

Table 2 Simulation parameters

Parameter	Value	Parameter	Value
M	5–30	P_m	100 mW
N	4	τ_m	1 s
J	6	σ^2	− 75 dBm
K	5–30	ξ_m	[1-2] Mbits
W_0	[1–2] MHz	f_m^0	[0.8–1.0] Gigacycles/s
$W_n^{\max}$	3	f_n^{m}	[10–20] Gigacycles/s
Dist	300 m	f_j^{d}	[1.2–1.5] Gigacycles/s
η_m	[5–6] Gigacycles	Path loss exponent	3

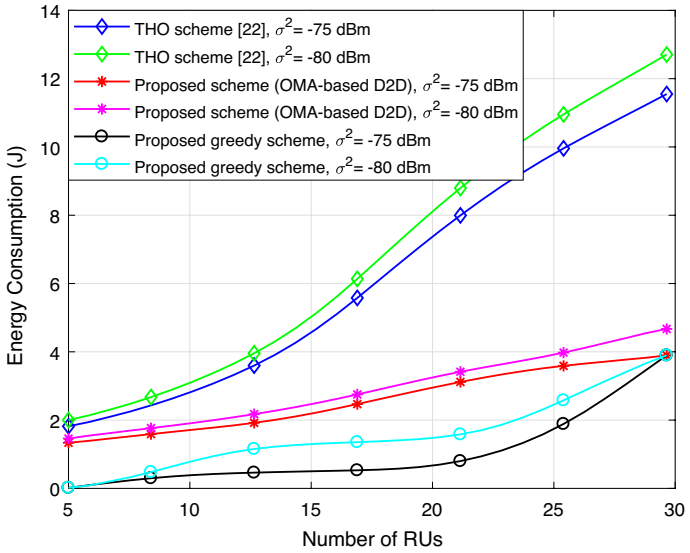

Fig. 2 Energy consumption versus the number of RUs

offloading mode is applied and no NOMA-based scheme is utilized. Furthermore, the performance of the algorithm proposed in [22] is also evaluated.

It can be observed from the figure that with the increase in the number of RUs, energy consumption increases as well. This is because as large number of RUs offload their tasks, the energy consumption due to task transmission and execution increase accordingly. In addition, it can also be seen that when noise power increases, the energy consumption also increases, the reason is that higher noise power leads to lower data transmission rate and longer time for task transmission, hence, higher energy consumption is resulted. Comparing the performance of the three algorithms, we can see that our proposed algorithm offers the lowest energy consumption which is benefited from the joint optimization of task offloading and resource allocation, as well as the performance gain of NOMA-based scheme.

In Fig. 3, we show the comparison results of the energy consumption versus the CPU cycles required for three different algorithms, i.e., proposed greedy scheme, THO scheme [22] and scheme proposed in [3]. Since the device execution efficiency can be examined by its processor's clock cycles (frequency), as lengthy instructions (or data) take more cycles to process as compared to short instructions. Therefore, there exists a direct relation between energy consumption and the number of CPU cycles required. The increase in the required CPU cycles indicates the higher complexity required to process the tasks, hence, higher energy consumption is required. It can also be observed that our proposed scheme outperforms the two comparative schemes.

Figure 4 shows the energy consumption versus the capacity of the MEC servers for the proposed scheme and the schemes proposed in [22] and [3]. It can be seen from the figure that with increasing the MEC server capacity by keeping the fixed task data size lowers the energy consumption due to the fact that the processor's clock frequency inversely affects the performance, as more clock cycles are available

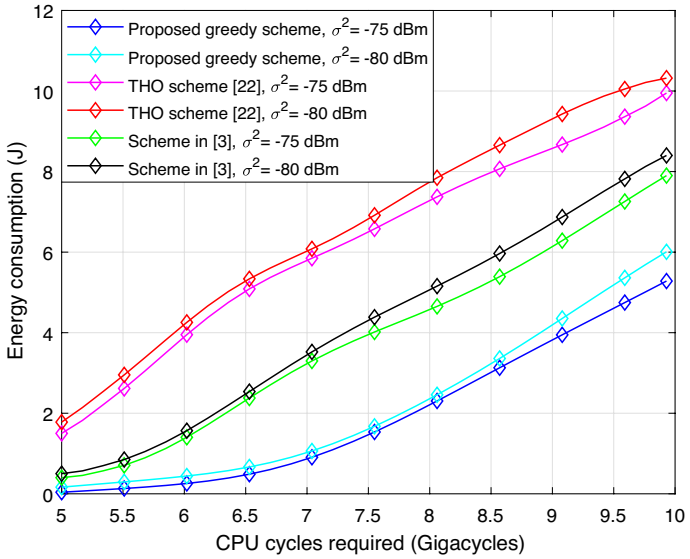

Fig. 3 Energy consumption versus the required CPU cycles

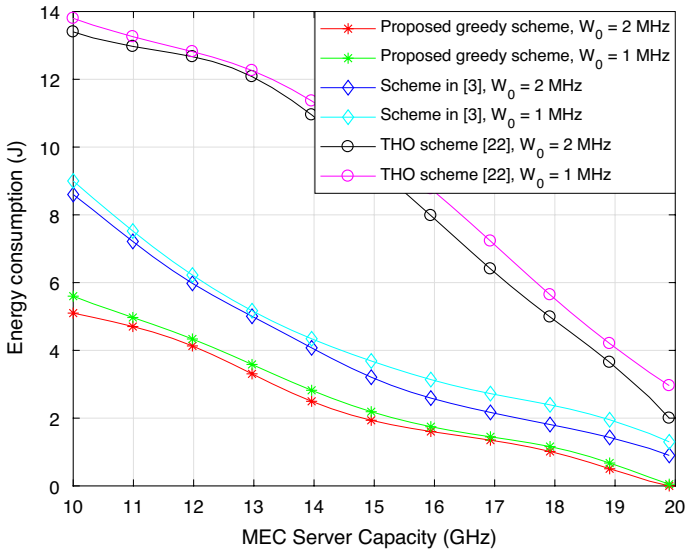

Fig. 4 Energy consumption versus the capacity of MEC servers

for fixed length data. The algorithms are evaluated against two different sub-channel bandwidth settings, i.e., $W_0 = 2$ MHz and $W_0 = 1$ MHz, and stating the fact that high bandwidth resource produces high information transmission rate and in turn lower energy consumption is produced. It can be observed from the figure that the proposed greedy scheme which integrates both OMA and NOMA schemes, outperforms the schemes proposed in [22] and [3] as more RUs prefer to offload their tasks to PUs as compared to local execution or offloading at far distant placed MEC servers.

In Fig. 5, an evaluation of energy consumption and task data size with different noise variances and sub-channel bandwidth settings is conducted for the proposed scheme and the schemes proposed in [22] and [3]. We can see that as the noise power

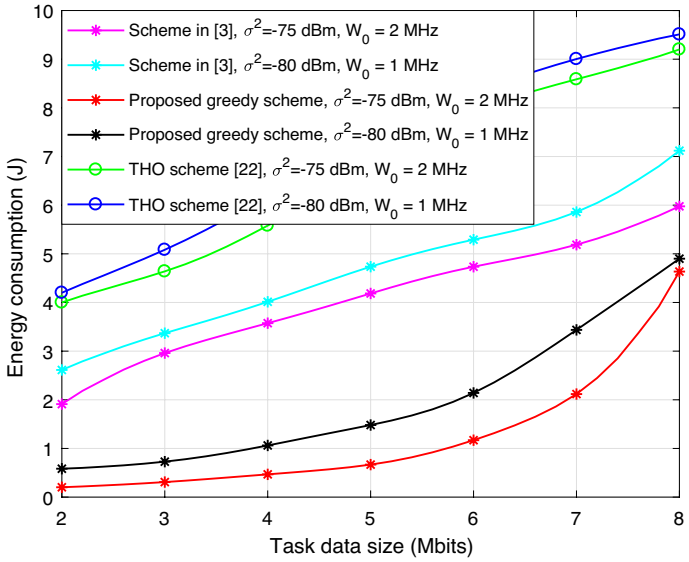

Fig. 5 Energy consumption versus the data size of tasks (different bandwidth)

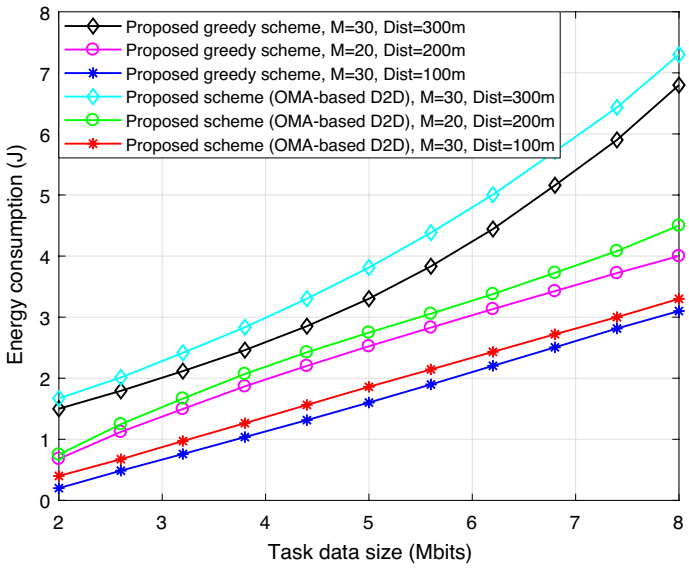

Fig. 6 Energy consumption versus the data size of tasks (different number of RUs)

increases, higher energy consumption is resulted. This is because increasing noise values decreases the signal-to-noise ratio (SNR), hence increasing energy consumption. From the figure, it can be noticed that with the rise of task data sizes, the energy consumption values tend to increase gradually. It can also be observed that our proposed scheme outperforms the schemes under comparison.

An illustration of the energy consumption versus the task data size over different D2D distance combinations and RU counts is provided in Fig. 6. According to the figure, when more RUs offload their tasks at large distances, we get higher energy consumption. This is due to the fact that long distance produces low data transmission rates, which ultimately leads to high transmission energy consumption in comparison

with short distances. Moreover, the proposed NOMA and OMA integrated algorithm outperforms the proposed OMA-based D2D scheme because the integrated algorithm yields better transmission performance, and lower energy consumption in turn.

7 Discussion

In this section, we will briefly discuss the computational complexity and convergence analysis of the proposed algorithm.

7.1 Computational complexity

In this section, the computation complexity of the proposed algorithm is analyzed. As the formulated problem is tackled according to two different use cases: no NOMA scheme and NOMA scheme, therefore, we examine the complexity of solving the both sub-problems, i.e., resource allocation sub-problem and computation offloading sub-problem according to use cases.

For the case where no NOMA scheme is applied, we virtualize the set of sub-channels into $N + J$ sets, the complexity of the K-M algorithm is $O\,(G^3)$ with $(11G^3{+}12G^2 {+}31G)/6$ maximum number of operations, where $G{=}N + 2J + 1$ [38].

For the case where NOMA scheme is applied, let K denote the RUs pairs in NOMA, the required number of operations needed using extensive search method is $M(J + 1)K$, having the computational complexity $O(M(J + 1)K)$. Figure 7 plots the computational complexity of the proposed greedy scheme.

7.2 Convergence analysis

It should be mentioned that through the process of algorithm execution, we conduct various sub-algorithms successively. Specifically, the sub-algorithms include: Sub-algorithm 1: determining local computing mode, Sub-algorithm 2: K-M algorithm-based

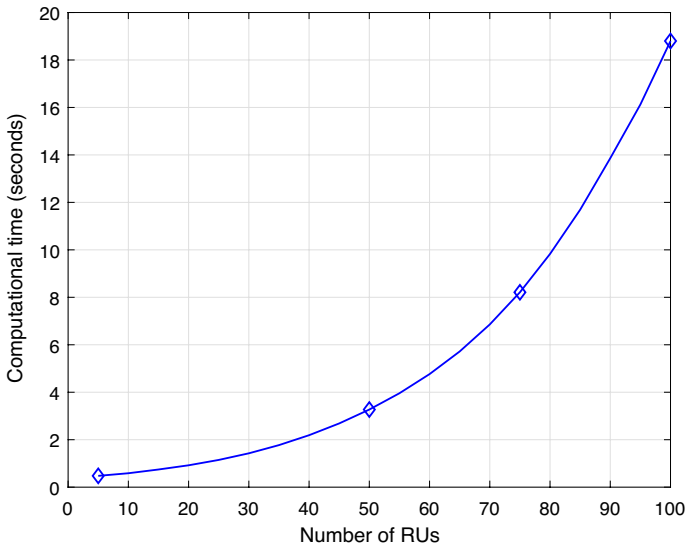

Fig. 7 Computational complexity of the proposed algorithm

sub-channel allocation, Sub-algorithm 3: priority-based sub-channel allocation for conflicting RUs, Sub-algorithm 4: greedy method-based task offloading and user pairing algorithm.

As Sub-algorithm 1 is conducted in an extensive manner and no iteration is required, the convergence can be reached easily. Sub-algorithm 2 is conducted in a centralized and noniterative mode, the strategy can be obtained directly by running the algorithm and the convergence of the algorithm is guaranteed. Sub-algorithm 3 is conducted iteratively. In each iteration, at least one RU with the highest priority is selected. Given the number of conflicting RUs, the number of RUs with the highest priorities is highly limited, which is in general much smaller than the number of conflicting RUs, hence, the algorithm convergence can be guaranteed, and the maximum iteration number can simply be set as the number of conflicting RUs.

Sub-algorithm 4 is applied to the RUs choosing D2D mode. Specifically, for various D2D pairs, the sub-algorithm is conducted independently. For an individual D2D pair, the energy consumption in OFDMA-based scheme and NOMA-based scheme is evaluated and the one offering better performance is selected. Hence, the algorithm convergence is guaranteed.

8 Conclusion

This paper jointly considers the computation offloading and resource allocation problem in a D2D-assisted and NOMA-empowered MEC systems. The original problem has been formulated as an energy consumption minimization problem that is NP hard; therefore, we have decomposed it into two sub-problems, i.e., resource allocation sub-problem and computation offloading sub-problem, and proposed two heuristic algorithms to obtain appropriate strategies for resource allocation and computation offloading. Numerical results have validated the effectiveness of the proposed scheme when compared with the relevant schemes [3, 22]. Future strategies might include extending the proposed scheme into an integrated network, e.g., the integration of satellites, unmanned aerial vehicles (UAVs) and cellular systems, which would utilize satellites and UAVs as MEC servers, so as to increase the flexibility and efficiency of task offloading. In addition, the task offloading strategy under dynamic scenarios with randomly arriving tasks and dynamically-changing channel models can also be investigated.

Abbreviations

MEC	Multi-access edge computing
D2D	Device-to-device
NOMA	Non-orthogonal multiple access
MD	Mobile device
OMA	Orthogonal multiple access
QoE	Quality of experience
VR	Virtual-reality
AR	Augmented-reality
PSO	Particle swarm optimization
MDP	Markov decision process
SAC	Soft actor critic
ADMM	Alternating direction method of multipliers
CCP	Convex–concave procedure
CPU	Central processing unit
THO	Two-stage heuristic optimization
5G	Fifth generation
SC	Superposition coding

SIC	Successive interference cancellation
QoS	Quality of service
CQR	Channel quality ranking
DRL	Deep reinforcement learning
VEC	Vehicular edge computing
EH	Energy harvesting
IoT	Internet of things
RU	Request user
PU	Providing user
K-M	Kuhn–munkres
SNR	Signal-to-noise ratio
UAV	Unmanned aerial vehicle

Acknowledgements
Not applicable.

Author contributions
UK proposes the system model, RC formulates it and assists with writing. UK simulates the proposed design and drafts the entire document in latex. All authors read and approved the final manuscript.

Funding
This work is supported by National Natural Science Foundation of China under Grant No. 62071078.

Availability of data and materials
Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 29 August 2022 Accepted: 20 December 2022
Published online: 10 January 2023

References

1. S. Sukhmani, M. Sadeghi, M. Erol-Kantarci, A. El Saddik, Edge caching and computing in 5G for mobile AR/VR and tactile internet. IEEE Multimed. **26**(1), 21–30 (2018)
2. Y. Zhao, W. Wang, Y. Li, C.C. Meixner, M. Tornatore, J. Zhang, Edge computing and networking: a survey on infrastructures and applications. IEEE Access **7**, 101213–101230 (2019)
3. J. Chen, Z. Chang, X. Guo, R. Li, Z. Han, T. Hämäläinen, Resource allocation and computation offloading for multi-access edge computing with fronthaul and backhaul constraints. IEEE Trans. Veh. Technol. **70**(8), 8037–8049 (2021)
4. C. Sun, X. Wu, X. Li, Q. Fan, J. Wen, V.C. Leung, Cooperative computation offloading for multi-access edge computing in 6g mobile networks via soft actor critic, in *IEEE Transactions on Network Science and Engineering* (2021)
5. P.A. Apostolopoulos, E.E. Tsiropoulou, S. Papavassiliou, Cognitive data offloading in mobile edge computing for internet of things. IEEE Access **8**, 55736–55749 (2020)
6. W. Wen, Y. Cui, T.Q. Quek, F.-C. Zheng, S. Jin, Joint optimal software caching, computation offloading and communications resource allocation for mobile edge computing. IEEE Trans. Veh. Technol. **69**(7), 7879–7894 (2020)
7. F. Guo, H. Zhang, H. Ji, X. Li, V.C. Leung, An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing. IEEE/ACM Trans. Netw. **26**(6), 2651–2664 (2018)
8. S. Yu, R. Langar, X. Fu, L. Wang, Z. Han, Computation offloading with data caching enhancement for mobile edge computing. IEEE Trans. Veh. Technol. **67**(11), 11098–11112 (2018)
9. H. Guo, J. Liu, Collaborative computation offloading for multiaccess edge computing over fiber-wireless networks. IEEE Trans. Veh. Technol. **67**(5), 4514–4526 (2018)
10. D. Chatzopoulos, C. Bermejo, E. ul Haq, Y. Li, P. Hui, D2d task offloading: a dataset-based Q &A. IEEE Commun. Mag. **57**(2), 102–107 (2018)
11. L. Zhang, J. Xu, Differential security game in heterogeneous device-to-device offloading network under epidemic risks. IEEE Trans. Netw. Sci. Eng. **7**, 1852–1861 (2019)
12. H. Long, C. Xu, G. Zheng, Y. Sheng, Socially-aware energy-efficient task partial offloading in MEC networks with d2d collaboration. IEEE Trans. Green Commun. Netw. **6**, 1889–1902 (2022)
13. M. Sun, X. Xu, X. Tao, P. Zhang, Large-scale user-assisted multi-task online offloading for latency reduction in D2D-enabled heterogeneous networks. IEEE Trans. Netw. Sci. Eng. **7**(4), 2456–2467 (2020)
14. Y. He, J. Ren, G. Yu, Y. Cai, D2d communications meet mobile edge computing for enhanced computation capacity in cellular networks. IEEE Trans. Wireless Commun. **18**(3), 1750–1763 (2019)
15. L. Pu, X. Chen, J. Xu, X. Fu, D2D fogging: an energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration. IEEE J. Sel. Areas Commun. **34**(12), 3887–3901 (2016)
16. Q. Chen, Z. Kuang, L. Zhao, Multiuser computation offloading and resource allocation for cloud-edge heterogeneous network. IEEE Internet Things J. **9**(5), 3799–3811 (2021)
17. U. Saleem, Y. Liu, S. Jangsher, X. Tao, Y. Li, Latency minimization for D2D-enabled partial computation offloading in mobile edge computing. IEEE Trans. Veh. Technol. **69**(4), 4472–4486 (2020)

18. M. Sun, X. Xu, Y. Huang, Q. Wu, X. Tao, P. Zhang, Resource management for computation offloading in D2D-aided wireless powered mobile-edge computing networks. IEEE Internet Things J. **8**, 8005–8020 (2020)
19. U. Saleem, Y. Liu, S. Jangsher, Y. Li, T. Jiang, Mobility-aware joint task scheduling and resource allocation for cooperative mobile edge computing. IEEE Trans. Wirel. Commun. **20**, 360–374 (2020)
20. X. Diao, J. Zheng, Y. Wu, Y. Cai, Joint computing resource, power, and channel allocations for D2D-assisted and NOMA-based mobile edge computing. IEEE Access **7**, 9243–9257 (2019)
21. I.A. Elgendy, W. Zhang, Y.-C. Tian, K. Li, Resource allocation and computation offloading with data security for mobile edge computing. Futur. Gener. Comput. Syst. **100**, 531–541 (2019)
22. H. Li, H. Xu, C. Zhou, X. Lü, Z. Han, Joint optimization strategy of computation offloading and resource allocation in multi-access edge computing environment. IEEE Trans. Veh. Technol. **69**(9), 10214–10226 (2020)
23. A.Y. Kiani, S.A. Hassan, B. Su, H. Pervaiz, Q. Ni, Minimizing the transaction time difference for NOMA-based mobile edge computing. IEEE Commun. Lett. **24**(4), 853–857 (2020)
24. Y. Huang, Y. Liu, F. Chen, Noma-aided mobile edge computing via user cooperation. IEEE Trans. Commun. **68**(4), 2221–2235 (2020)
25. Y. Wu, K. Ni, C. Zhang, L.P. Qian, D.H. Tsang, Noma-assisted multi-access mobile edge computing: a joint optimization of computation offloading and time allocation. IEEE Trans. Veh. Technol. **67**(12), 12244–12258 (2018)
26. Z. Wan, D. Xu, D. Xu, I. Ahmad, Joint computation offloading and resource allocation for NOMA-based multi-access mobile edge computing systems. Comput. Netw. **196**, 108256 (2021)
27. Z. Ding, D. Xu, R. Schober, H.V. Poor, Hybrid NOMA offloading in multi-user MEC networks. IEEE Trans. Wirel. Commun. **21**, 5377–5391 (2022)
28. B. Zhu, K. Chi, J. Liu, K. Yu, S. Mumtaz, Efficient offloading for minimizing task computation delay of NOMA-based multiaccess edge computing. IEEE Trans. Commun. **70**(5), 3186–3203 (2022)
29. I. Altin, M. Akar, A joint resource allocation method for hybrid NOMA MEC offloading. Phys. Commun. **54**, 101809 (2022)
30. J. Du, Y. Sun, N. Zhang, Z. Xiong, A. Sun, Z. Ding, Cost-effective task offloading in NOMA-enabled vehicular mobile edge computing. IEEE Syst. J. 1–12 (2022). https://doi.org/10.1109/JSYST.2022.3167901
31. C. Zheng, W. Zhou, Computation bits maximization in backscatter-assisted wireless-powered NOMA-MEC networks. EURASIP J. Wirel. Commun. Netw. **2022**(1), 1–21 (2022)
32. L. Qian, Y. Wu, J. Ouyang, Z. Shi, B. Lin, W. Jia, Latency optimization for cellular assisted mobile edge computing via non-orthogonal multiple access. IEEE Trans. Veh. Technol. **69**(5), 5494–5507 (2020)
33. H. Lin, Y. Cao, Y. Zhong, P. Liu, Secure computation efficiency maximization in NOMA-enabled mobile edge computing networks. IEEE Access **7**, 87504–87512 (2019)
34. Z. Ding, P. Fan, H.V. Poor, Impact of non-orthogonal multiple access on the offloading of mobile edge computing. IEEE Trans. Commun. **67**(1), 375–390 (2018)
35. L.P. Qian, B. Shi, Y. Wu, B. Sun, D.H. Tsang, Noma-enabled mobile edge computing for internet of things via joint communication and computation resource allocations. IEEE Internet Things J. **7**(1), 718–733 (2019)
36. Z. Song, Y. Liu, X. Sun, Joint task offloading and resource allocation for NOMA-enabled multi-access mobile edge computing. IEEE Trans. Commun. **69**(3), 1548–1564 (2020)
37. M.S. Ali, H. Tabassum, E. Hossain, Dynamic user clustering and power allocation for uplink and downlink non-orthogonal multiple access (NOMA) systems. IEEE Access **4**, 6325–6343 (2016)
38. J. Munkres, Algorithms for the assignment and transportation problems. J. Soc. Ind. Appl. Math. **5**(1), 32–38 (1957)

Publisher's Note