Dynamic Request Scheduling Optimization in Mobile Edge Computing for IoT Applications

Shihong Hu and Guanghui Li

***Abstract*—In the era of 5G, with the increasing demands on computation and massive data traffic of the Internet of Things (IoT), mobile edge computing (MEC) and ultradense network (UDN) are considered to be two enabling and promising technologies, which result in the so-called ultradense edge computing (UDEC). Task offloading as an effective solution offers low latency and flexible computation for mobile users in the UDEC network. However, the limited computing resources at the edge clouds and the dynamic demands of mobile users make it challenging to schedule computing requests to appropriate edge clouds. To this end, we first formulate the transmitting power allocation (PA) problem for mobile users to minimize energy consumption. Using the quasiconvex technique, we address the PA problem and present a noncooperative game model based on subgradient (NCGG). Then, we model the problem of joint request offloading and resource scheduling (JRORS) as a mixed-integer nonlinear program to minimize the response delay of requests. The JRORS problem can be divided into two problems, namely, the request offloading (RO) problem and the computing resource scheduling (RS) problem. Therefore, we analyze the JRORS problem as a double decision-making problem and propose a multiple-objective optimization algorithm based on i-NSGA-II, referred to as MO-NSGA. The simulation results show that NCGG can save the transmitting energy consumption and has a good convergence property, and MO-NSGA outperforms the existing approaches in terms of response rate and can maintain a good performance in a dynamic UDEC network.**

***Index Terms*—Internet of Things (IoT), mobile edge computing (MEC), optimization, resource scheduling (RS), ultradense network (UDN).**

I. Introduction

With the wide application of wireless communication technology and the rich variety of sensors, mobile devices in the Internet of Things (IoT), such as smart cars, mobile phones, and unmanned aerial vehicles, can access the Internet via a cellular network or a low-power wide-area network, etc. The explosive surge of mobile application types has placed stringent demands on computation capacity and real-time processing. A large number of intensive computing requests are bound to accelerate the energy consumption of mobile devices and shorten their lifetime. In particular, mobile AR devices enhance the real world by rendering virtual overlays on the user's field of view based on the understanding of the surroundings through the camera [1], [2]. However, most existing AR systems can detect the surfaces but lack the sufficient computing capacity to detect and classify complex objects in the world. Offloading the computation to the cloud is challenging due to the stringent requirements on high accuracy and low latency. To this end, mobile edge computing (MEC) [3] as a promising technology can effectively overcome the shortcomings of traditional mobile cloud computing. Mobile network operators and cloud service providers provide rich communication and computing resources at the edge of the network, including base stations (BSs) and local wireless access points (APs) in a cooperative manner. Mobile devices can acquire the computing resources and services at close proximity to the edge network via high-speed wireless access networks [4]. In light of this, using edge computing to offload computation requests brings ultralow latency and flexible computing for computation-intensive requests from mobile devices.

A. Motivation

Cisco [5] reported that the global mobile data traffic will increase sevenfold in the next five years, while the number of global mobile devices will be 12.3 billion by 2022. This will make it difficult for mobile devices to access the network to obtain the computing resources of the edge clouds. Therefore, the critical problem is how to optimally schedule the computing resources of MEC to the competing demands from mobile devices. In recent years, this problem has been received significant attention, and the existing works [6]–[9] only considered a request/task can be offloaded to the edge cloud while the local execution fails to accomplish the computation. However, these works ignore the fact that the limited computing resources of the edge cloud may consume more energy and bring additional delays, as the offloaded request/task with a large amount of data or high workload. In addition, Tran and Pompili [10] solved the joint task offloading and resource allocation problem with orthogonal frequency-division multiple access (OFDMA) as

Manuscript received September 14, 2019; revised November 16, 2019; accepted November 19, 2019. Date of publication November 22, 2019; date of current version February 11, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61472368, in part by the Key Project of the Jiangsu Provincial Research and Development under Grant BE2016627, in part by the Wuxi International Science and Technology Research and Development Cooperative Project under Grant CZE02H1706, and in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX_1862. *(Corresponding author: Guanghui Li.)*

S. Hu is with the School of IoT Engineering, Jiangnan University, Wuxi 214122, China (e-mail: jnuhsh@163.com).

G. Li is with the School of IoT Engineering, Jiangnan University, Wuxi 214122, China, and also with the Research Center for IoT Technology Application Engineering (MOE), Jiangnan University, Wuxi 214122, China (e-mail: ghli@jiangnan.edu.cn).

Digital Object Identifier 10.1109/JIOT.2019.2955311

the multiple access scheme. However, mobile devices cannot use the entire system bandwidth simultaneously to transmit data in the OFDMA scheme, which may lead to network congestion and increase the energy consumption of mobile devices. Mao *et al.* [11] and Shojafar *et al.* [12] have studied the joint computation offloading and resource scheduling (RS) problem, however, they only considered a single BS (i.e., a macro-BS) to provide access services for IoT devices.

To cope with the massive device connections and data traffic, a multibase station collaborative service scenario, an ultradense network (UDN) [13], [14] under the 5G architecture, has gradually become widely accepted by the mobile network operators. In a UDN, the mobile network operator deploys a large number of micro-BSs and macro-BSs to provide services for mobile devices together. Each micro-BS is associated with the edge cloud via a local area network, and the macro-BS is associated with the resource-rich deep cloud through the Internet. Mobile devices in a UDN can select to offload computation requests to the macro-BS when the micro-BSs are not able to process all offloaded requests. A major limitation in UDN is that each micro-BS with a single edge cloud providing access services for mobile devices is much more computationally intensive than macro-BS. Moreover, for a large number of offloading requests, the limited computing resources at the edge cloud in an edge-computing system may increase the request–response latency. Therefore, an appropriate two-way offloading scheme is worth studying. The nonorthogonal multiple access (NOMA) protocol as a promising radio access technology for 5G system has attracted extensive attention [15]–[17]. The NOMA allows all mobile devices to use the entire system bandwidth simultaneously to transmit data, but it has problems with multiple access interference (MAI).

We present a dynamic request scheduling scheme in MEC for IoT applications using the concept of UDN in this article. In an ultradense edge computing (UDEC) environment, we assume that all mobile devices, including vehicles and intelligent terminals, are mobile users. Different from the previous work, we consider a scenario where computation requests from mobile users can be offloaded to both macro-BS and micro-BS. Mobile users in the same zone are supposed to be associated with the micro-BS that covered the zone, and the macro-BS is assumed to be associated with all mobile users in an UDEC network. Due to the demands of requests are dynamic and the mobility of users, the proposed solution can be flexible to adapt to the dynamic system. Especially, considering the MAI among the mobile users in the uplink channel between users and BSs by applying NOMA, we also propose an uplink power allocation (PA) algorithm for mobile users to minimize the transmitting energy consumption.

B. Contribution

In the context of UDN, we consider an UDEC network consisting of a macro-BS, many micro-BSs, and a large number of mobile users under the 5G architecture. The multiple access scheme between users and BSs is the NOMA protocol. To take full advantage of the benefits of the request offloading (RO) in the considered system, we should address several critical challenges. First, the PA problem is challenging to solve because the interference among different mobile users affects the uplink rate, which makes the problem unconvex. Second, the RO decision is hard to make because each mobile user needs to decide to offload the computing request to its associated micro-BS or the macro-BS, and the result of offloading decision directly affects the RS strategy of the associated micro-BS. Third, the computing RS policy is not only affected by RO but also influenced by the variability of the request profiles from mobile users. The main contributions of this article can be summarized as follows.

1) We formulate a PA problem for mobile users to minimize the transmitting energy consumption, and the PA problem can be addressed using the quasiconvex technique. Then, we present a noncooperative game model based on subgradient (NCGG) for PA problem.
2) We model the problem of joint RO and RS (JRORS) as a mixed-integer nonlinear program to maximize the system welfare. The JRORS problem can be separated into two problems, namely, the RO problem and the computing RS problem. Therefore, we analyze the JRORS problem as a double decision-making problem, which is very complex and involves a tradeoff between two conflicting objectives.
3) We propose a multiple-objective optimization algorithm based on i-NSGA-II, referred to as MO-NSGA, to solve the JRORS problem. The MO-NSGA uses the idea of nondominated solutions and presents a novel crossover based on direction, which accelerates the optimization.
4) We conduct extensive experiments to evaluate the performance of the proposed algorithms. The experimental results show that NCGG can save the transmitting energy compared with other algorithms and have a good convergence property. The MO-NSGA always outperforms the existing approaches in terms of response rate and can maintain a good performance in a dynamic MEC system.

The remainder of this article is organized as follows. In Section II, we review the related works. In Section III, we describe the system model and formulate the problems. We give a detailed analysis of the problems and present our efficient algorithms in Section IV. Section V shows the results and discussions of experiments. Finally, we conclude this article in Section VI.

II. Related Work

The basic idea of the UDN is to make mobile users close to the BSs, which represents a novel paradigm in future networks. In recent years, UND has been extensively studied [18]–[21]. Kamel *et al.* [19] presented a survey on UDN, and they compared the recent works on many research directions, such as interference management, RS, and propagation modeling. Osseiran *et al.* [21] discussed the use of infrastructure densification by UDNs to meet the high traffic demands, which can increase the capacity and energy efficiency of radio links and make better use of the

spectrum. Similarly, López-Pérez *et al.* [20] brought further understanding and analyzed the potential gains of UDN paradigms. Yu *et al.* [22] proposed a novel MEC framework in UDNs for IoT applications. In the framework, the macro-BS is used as the central controller to schedule task offloading, BS sleeping, and user-base association. The task engine process is executed on the macro-BS to collect task information, computing resource information of edge clouds in micro-BSs, and the network status. To improve the stability of the association and the energy efficiency of the system, Ma *et al.* [23] introduced an efficient user association scheme using robust optimization. Therefore, we adopt a UDN architecture in MEC network considering the advantages of UDN in the rich capacity of radio links for mobile users.

Considerable researches have been devoted to task offloading and resource allocation in the MEC network [24]–[30]. Bahreini *et al.* [31] developed an auction-based scheme that allocates and prices the edge resources in the MEC network. Mobile users with heterogeneous demands place bids to compete for the resources of the edge cloud. By combining the features of both positions and combinatorial auctions, the proposed scheme efficiently handles the resource allocation problem. Similarly, using an auction theory, the work in [28] proposed a two-time scale scheme to allocate the computing and communication resources in the hierarchical MEC network. The authors established the profit of the system by offering resources to mobile users, to formulate the auction-based model aiming at maximizing the system profit. However, both [28] and [31] did not consider the computation association of mobile users; they all assumed that the computation from mobile users had been offloaded to the appropriate edge cloud. Tran and Pompili [10] formulated the problem involved jointly optimizing the task offloading policy, transmitting power of mobile users, and the resource allocation at the edge servers. They decomposed the problem into two independent problems, including resource allocation and task offloading. Using the quasiconvex and convex techniques, they addressed the problems efficiently. Misra and Saha [32] proposed a task offloading scheme for IoT applications in soft-defined access networks, where devices are connected to edge clouds by multihop paths. They developed the dynamic task offloading problem as a nonlinear optimization program under the constraints of IoT devices and dynamic network status. To address the problem, they adopted the linearization approach to transform the problem into an integer linear program. The proposed greedy solution can reduce the average delay and energy consumption efficiently. However, the existing task offloading schemes did not consider the dynamic network status in the MEC system.

In recent years, the task offloading problem in a UDEC network has attracted wide attention. Chen and Hao [7] formulated the task offloading problem for a UDEC network to minimize the delay while reducing the energy consumption of mobile users. They proposed an innovative framework for task offloading, by deploying controller at macro-BS to obtain the global information about mobile users, BSs, and tasks. Guo *et al.* [33] presented a definition for the computation offloading problem for a UDEC network and proposed a heuristic greedy scheme to solve the problem. The computation resources, the MEC servers, and mobile users are utilized collaboratively in the proposed scheme. In the later work [34], Guo *et al.* pointed out that the existing works on task offloading for UDEC networks ignored the different types of requests from mobile users. Then, they further studied the multiuser task offloading problem with multitype requests. To address the problem, they proposed a game-based joint offloading scheme. However, most of the existing works on task offloading for MEC using the concept of UND did not take into account the influence of communication between mobile users and BSs and the computing capacity of edge clouds. In this article, we address the dynamic request scheduling problem involving jointly optimizing the transmitting PA policy, the RO for mobile users and the computing RS at the edge clouds.

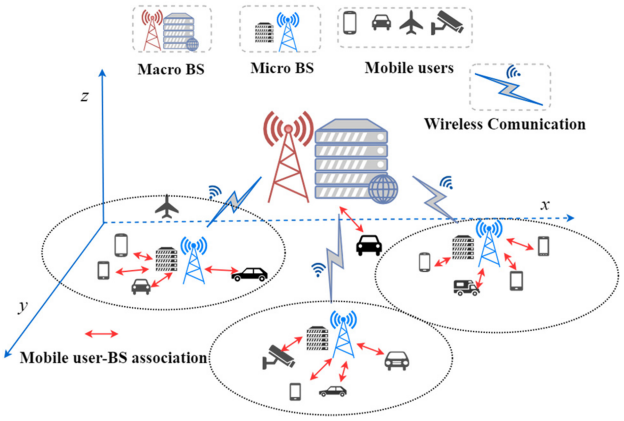

Fig. 1. System model.

III. System Model

As shown in Fig. 1, we consider a UDEC network consisting of a set of mobile users, U, a set of micro-BSs (micro-BSs are abbreviated as BSs in the following) with edge servers, N, and a macro-BS with a deep cloud, C. It is assumed that each BS covers a local area called a zone, and a mobile user should be associated with only one zone. Edge server may be a physical server or a virtual machine with computing capacities, and we assume that its associated BS is interconnected by the backhaul links, allowing a mobile user to be served by a nonlocal BS. Each mobile user can offload computing request to a BS in its zone. Like [22], we assume that the macro-BS is used as the central controller, which is responsible for collecting task information, computing resource information of edge clouds in BSs, and the network status. Specifically, the set of mobile users and BSs is denoted by $U = \{1, 2, \ldots, u\}$ and $N = \{1, 2, \ldots, n\}$, respectively. We assume each mobile user $u \in U$ generate one computing request at a time, given as $q_u = <w_q, s_q, pr_q, Tg_q, Tb_q>$. Here, w_q denotes the workload of request q, i.e., the required computing to accomplish the request, and s_q denotes the request input data size. We use pr_q to denote the request priority representing the importance of different requests. Tg_q and Tb_q are ideal delay and tolerable

TABLE I
KEY NOTATIONS

Symbol	Description
U	Set of mobile users
N	Set of BSs equipped with edge servers
Q	Set of requests generated by mobile users
B	The fixed bandwidth
H	The fixed altitude of BS
σ_o^2	The background white Gaussian noise power
q_u	Computing request of user u
w_q	Workload of request q
pr_q	Priority of request q
I_q	Input data of request q
Tg_q	Ideal delay of request q
Tb_q	Tolerable delay of request q
p_{max}	The maximum transmitting power of mobile user
p_{un}	The transmitting power from user u to BS n
P_{BS}	The average power consumption of BS
P_C	The average power consumption of macro-BS
R_n	Computing capacity of BS (edge server) n
R_c	Computing capacity of micro-BS with a deep cloud server C
R_{qn}	Computing resources that BS n schedules to request q
$x_{qn} \in \{0, 1\}$	Indicator of allocating request q to BS n
P	Power allocation policy
X	Request offloading policy
Y	Computing resource scheduling policy
v_{up}	uplink rate from user to BS
t_{up}^q	Uplink transmission time of request q to BS n
t_{pro}^q	Process time of request q at BS
t_q	Total response time of request q
e_{qn}	Total energy consumption of finishing request q at BS n

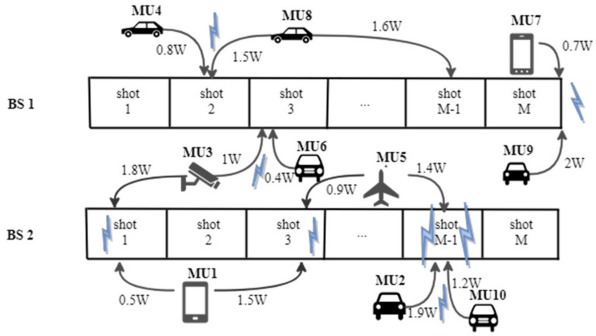

Fig. 2. Illustration of the NOMA protocol.

delay thresholds, respectively. Considering that the position of mobile user varies over time, we use $p_u^t = (x_u, y_u, 0)$ to denote the location of mobile user u at time t. All BSs are fixed and the location of BS n is given as $p_n^t = (x_n, y_n, H)$ with the same attitude h. For ease of reference, Table I summarizes the key notations.

A. Delay Model

In this article, we apply NOMA as the communication scheme between mobile users and BSs as illustrated in Fig. 2. Therefore, the mobile users in the same zone can transmit data to BS simultaneously at the expense of the interference. In this case, the interference may cause performance degradation, i.e., the decrease of uplink rate. We define the transmitting PA policy as $p = \{p_{un} | u \in U, q \in Q\}$, in which p_{un} denotes the transmitting power from mobile user u to BS n. The location of each mobile user is assumed to be unchanged during the time interval, and the uplink rate $v_{un}(t)$ from mobile user u to BS n can be formulated as follows:

$$v_{un}(t) = B \log_2 \left(1 + \frac{p_{un}(t) g_{un}(t)}{\sigma_o^2 + \sum_{\substack{u' \\ u' \neq u}}^{U_n} p_{u'n}(t) g_{u'n}(t)} \right) \tag{1}$$

where B and σ_o^2 represent the bandwidth of the uplink system and background white Gaussian noise power, respectively. The channel power gain [35] between mobile user u to BS n is defined as follows:

$$g_{un}(t) = \frac{g_0}{(x_u - x_n(t))^2 + (y_u - y_n(t))^2 + H^2} \quad u \in U, n \in N, t \in T \tag{2}$$

where g_0 represents the channel power gain at the reference distance $d_0 = 1$ m and the transmitting power is 1 W. We define the RO policy as $X = \{x_{qn} | q \in Q, n \in N\}$, in which x_{qn} is a binary variable and $x_{qn} = 1$ indicates that request q is offloaded to BS n, and $x_{qn} = 0$ indicates the request q is offloaded to macro-BS. Thus, the time taken to transmit data I_q from mobile user u for offloading is given as

$$t_{\text{up}}^q = \begin{cases} \frac{I_q}{v_{un}(t)}, & x_{qn} = 1 \\ \frac{I_q}{v_{un}(t)}, & x_{qn} = 0. \end{cases} \tag{3}$$

We define the computing RS policy as $Y = \{R_{qn} | q \in Q, n \in N\}$, in which R_{qn} denotes the amount of computing resource that BS n schedules to request q. Thus, the execution time of request q at BS or macro-BS is given as

$$t_{\text{pro}}^q = \begin{cases} \frac{l_q}{R_{qn}}, & x_{qn} = 1 \\ \frac{l_q}{R_c}, & x_{qn} = 0 \end{cases} \tag{4}$$

where R_c is the computing capacity of macro-BS. Therefore, we obtain the total delay for offloading request q

$$t_q = t_{\text{up}}^q + t_{\text{pro}}^q. \tag{5}$$

B. Energy Model

The energy consumption for offloading requests includes the energy consumed for transmitting the data and the energy consumption of processing requests. Thus, we define the transmitting energy consumption for data offloading from mobile user u to BS n at time t as

$$E_u^{\text{tra}}(t) = p_{un}(t) t_{\text{up}}^q. \tag{6}$$

Given the average power consumption of BS and macro-BS, the energy consumed by executing request q is defined as

$$E_u^{\text{pro}}(t) = \begin{cases} p_{BS} t_{\text{pro}}^q, & x_{qn} = 1 \\ p_C t_{\text{pro,c}}^q, & x_{qn} = 0 \end{cases} \tag{7}$$

where P_{BS} and P_C are the average power consumption of BS and macro-BS.

C. Problem Formulation

We assume that each mobile user is selfish and makes efforts to minimize their energy consumption for transmitting data. The NOMA protocol is applied as the communication scheme in this article, so mobile users can transmit data simultaneously by using the entire system bandwidth. In this case, mobile users using larger transmitting power can reduce the transmission delay but may lead to more interference and energy consumption. To minimize the energy consumption for transmitting data of the whole system at time t, we formulate the PA problem as follows:

$$P1: \min_{P} \ E = \sum_{n}^{N} \sum_{u}^{U} E_u^{tra}(t) \tag{8a}$$

$$\text{s.t.} \ \ 0 \le p_{un}(t) \le p_{\max} \quad \forall n \in N, \forall u \in U. \tag{8b}$$

Objective (8a) minimizes the energy consumption for transmitting data with $E_{un}^{tra}(t)$ as given in (6). Specifically, constraint (8b) ensures that the transmitting power of each mobile user is less than $p_{\max}$ and greater than 0. Constraints (8c) and (8d) imply that each request generated by mobile users can be either offloaded to only one BS or macro-BS.

Obviously, the purpose of the mobile user is to reduce the response delay of offloading request to get an ideal result. In general, mobile users in the same zone compete for the computing resources of the same BS to complete the requests within the ideal delay. Referring to [36], we define the edge system utility for processing request q as

$$k_q = \begin{cases} 1, & t_q \le Tg_q \\ 1 - \frac{1}{1+e^{\alpha\left(T_{\text{avg}} - t_q\right)/\left(T_{\text{avg}} - Tg_q\right)}}, & Tg_q < t_q \le T_{\text{avg}} \\ \frac{1}{1+e^{\alpha\left(t_q - T_{\text{avg}}\right)/\left(Tb_q - T_{\text{avg}}\right)}}, & T_{\text{avg}} < t_q \le Tb_q \\ 0, & t_q > Tb_q \end{cases} \tag{9}$$

where

$$T_{\text{avg}} = \frac{Tg_q + Tb_q}{2} \tag{10}$$

and we define the edge system cost for processing request q as

$$c_q = \alpha \int_{E_0 - E_r^t}^{E_0 - E_r^t + E_u^{\text{pro}}} e^{x/10} dx \tag{11}$$

where α is a user-defined constant to ensure that c_q is in the range [0, 1], and E_0 and E_r^t are the initial energy and residual energy at time t of BS, respectively. With the increase of energy consumption of executing requests, the energy cost c_q of the edge server is increased. Given the fixed computing resources, the BS may not be able to process all requests in a timely manner. Therefore, the mobile users can choose to send the request to the macro-BS for processing, and the edge system should pay for this article. The extra cost for offloading to macro-BS is defined as

$$e_q = \varepsilon k_q + (1 - \varepsilon) E_q^{\text{pro}} \tag{12}$$

where ε is a constant implying the relative importance of total delay and executing energy consumption. Thus, we define the total system welfare as

$$W = \sum_{n}^{N} \sum_{q}^{Q} \left[x_{qn}\left(k_q - c_q\right) - \left(1 - x_{qn}\right) e_q \right]. \tag{13}$$

We formulate the JRORS problem as a system welfare maximization problem

$$P2: \max_{X,Y} \ W \tag{14a}$$

$$\text{s.t.} \ \sum_{n \in N} x_{qn} \le 1 \quad \forall q \in Q \tag{14b}$$

$$x_{qn} \in \{0, 1\} \quad \forall q \in Q, n \in N \tag{14c}$$

$$\sum_{q \in Q} R_{qn} \le R_n \quad \forall n \in N \tag{14d}$$

$$R_{qn} > 0, \quad \forall q \in Q, \ n \in N. \tag{14e}$$

Constraints (14b) and (14c) imply that each request generated by mobile users can be either offloaded to only one BS or macro-BS. Given the fixed computing resources, the BS may not be able to process all requests in a timely manner. Therefore, the mobile users can choose to send the request to the cloud center for processing. Constraint (14d) ensures that the total computing resources scheduled to requests should not exceed the BS's computing capacity. Constraint (14e) ensures that BS must schedule a positive computing resource to each request that offloaded to it.

IV. Efficient Algorithms

A. Power Allocation

Specifically, the PA problem can be expressed as

$$\frac{p_{un}(t) I_u}{v_{un}(t)} = \max_{P} \ E = \sum_{n}^{N} \sum_{u}^{U} \sum_{n}^{N} \sum_{u}^{U} \times \frac{p_{un}(t) I_u}{B \log_2 \left(1 + \frac{p_{un}(t) g_{un}(t)}{\sigma_o^2 + \sum_{\substack{u' \\ u' \ne u}}^{U_n} p_{u'n}(t) g_{u'n}(t)} \right)} \tag{15a}$$

$$\text{s.t.} \ \ 0 \le p_{un}(t) \le p_{\max} \quad \forall n \in N, \ \forall u \in U. \tag{15b}$$

Problem (15) is difficult to solve because the objective function in (15a) is nonlinear, and the term $v_{un}(t)$ depends on the transmitting power and location of the other mobile users in the same zone, as shown in (1) and (2). We assume that each BS calculates its associated mobile users' PA policy P_n independently to minimize the energy consumption E_n at each time t. Then, the PA problem can be solved by solving a set of subproblems as given as

$$\min_{P_n} \ E_n = \sum_{u}^{U_n} \Phi(p_{un}) = \sum_{u}^{U_n} \frac{p_{un} I_u}{B \log_2(1 + \gamma p_{un})} \tag{16a}$$

$$\text{s.t.} \ \ 0 \le p_{un}(t) \le p_{\max} \quad \forall u \in U_n \tag{16b}$$

where

$$\gamma = \frac{g_{un}(t)}{\sigma_o^2 + \sum_{\substack{u' \\ u' \ne u}}^{U_n} p_{u'n}(t) g_{u'n}(t)}. \tag{17}$$

Algorithm 1 NCGG

Initialize: $\boldsymbol{p^0} = (p_{max}, p_{max}, \ldots, p_{max})_{Un}, r = 0$
repeat
  $r = r + 1$
  $\mu_r = 0.1/\sqrt{r}$
  for each $u \in U_n$
    Calculate δ_u^r and p_u^r
    if $p_u^r < 0$ **then**
      $p_u^r = 0$
    end if
    $p_u^r = \arg\min E(p_u^r, p_{-u}^r)$
  end for
Until $\boldsymbol{p^r}$ is a NE

The second-order derivative of objective (16a) with respect to p_{un} is not always positive, so problem (16) is nonconvex. However, we use the quasiconvex technique to address the problem.

Theorem 1: The utility function (16a) is strictly quasiconvex in constraint (16b).

Proof: See the Appendix. ∎

As the utility function (16a) is quasiconvex and the feasible set constraint (16b) is convex, closed, and bounded, there exists at least one Nash equilibrium in $\Gamma = <U, S, F>$. Thus, we designed an NCGG for PA problem. The strategy profile of mobile user u is defined as $s_u, s_u = \{p_{un}\}$, and the strategy profiles of other mobile users are represented as $s_{-u}, s_{-u} = \{p_{-un} | -u \in U_n, -u \neq u\}$. Let $S = \{S_1, S_{2,\ldots,}S_{u,\ldots,}S_{Un}\}$ denotes the set of strategy profiles of all mobile users, in which S_u denotes the set of possible strategies of mobile user u. Let F denotes the set of mobile users' utility functions, i.e., $F = \{E(s_1, s_{-1}), E(s_2, s_{-2}), \ldots, E(s_u, s_{-u}), \ldots, E(s_{Un}, s_{-Un})\}$. The noncooperative game can be formulated as $\Gamma = <U, S, F>$, each mobile user as a player $u \in U_n$ in the game tries to minimize its utility function, i.e., the transmitting energy consumption. For each mobile user $u \in U_n$, its strategy set is discretized into a descending order set

$$S_u = \left\{p_u^{\max} = p_u^1, p_u^2, \ldots, p_u^\eta = p_u^{\min}\right\} \tag{18}$$

and the dynamic step size δ_u^r is based on the subgradients of $E(s_u, s_{-u})$ at each iteration r, which is given as

$$\delta_u^r = \mu_r \nabla\ E(s_i, s_{-i}) = \mu_r \frac{\partial E(s_i, s_{-i})}{\partial p_{un}} = \frac{\mu_r I_u (h(p_{un}) g(p_{un}) - \gamma p_{un})}{B g(p_{un}) h^2(p_{un})} \tag{19}$$

where μ_r is a step size parameter varies with the number of iterations. Therefore, we know that

$$p_u^r = p_u^{r-1} - \delta r. \tag{20}$$

Then, the procedure of NCGG is shown in Algorithm 1. Each mobile user in U_n initializes its transmitting power to $p_{\max}$ at the beginning. Before, the transmitting power of users make the game reach the Nash equilibrium, the game process will repeat. The step size μ_r is diminishing as iteration decreases. In each iteration, the mobile users update their transmitting power according to (20) to minimize the energy consumption.

B. Joint Request Offloading and Computing Resource Scheduling

After allocating the transmitting power of mobile users, the delay-sensitive requests from users need to be offloaded to BSs or macro-BS. Specifically, the JRORS problem can be expressed as

$$\max_{X,Y}\ W = \sum_n^N \sum_q^Q \Big[x_{qn}\big(k_q - c_q\big) - \big(1 - x_{qn}\big)\Big(\varepsilon k_q + (1-\varepsilon)E_q^{\text{pro}}\Big)\Big] \tag{21a}$$

$$\text{s.t.} \sum_{n \in N} x_{qn} \leq 1 \quad \forall q \in Q \tag{21b}$$

$$x_{qn} \in \{0,\ 1\} \quad \forall q \in Q, n \in N \tag{21c}$$

$$\sum_{q \in Q} R_{qn} \leq R_n \quad \forall n \in N \tag{21d}$$

$$R_{qn} > 0 \quad \forall q \in Q,\ n \in N. \tag{21e}$$

We observe that constraints (21b) and (21c) of the offloading policy X, and constraints (21d) and (21e) of the offloading policy Y are separated from each other. Problem (21) can be divided into two problems, namely, the RO problem and the computing RS problem. Hence, the RO problem of minimizing the extra cost of the edge system can be expressed as

$$\min_X\ M = \sum_n^N \sum_q^Q \big(1 - x_{qn}\big)\Big(\varepsilon k_q + (1-\varepsilon)E_q^{\text{pro}}\Big) \tag{22a}$$

$$\text{s.t.} \sum_{n \in N} x_{qn} \leq 1 \quad \forall q \in Q \tag{22b}$$

$$x_{qn} \in \{0,\ 1\} \quad \forall q \in Q, n \in N \tag{22c}$$

and the RS problem of maximizing the edge system welfare can be expressed as

$$\max_Y\ W = \sum_n^N \sum_q^Q \big[x_{qn}\big(k_q - c_q\big)\big] \tag{23a}$$

$$\sum_{q \in Q} R_{qn} \leq R_n \quad \forall n \in N \tag{23b}$$

$$R_{qn} > 0 \quad \forall q \in Q,\ n \in N. \tag{23c}$$

Therefore, the JRORS problem is a double decision-making problem which is very complex and involves a tradeoff between two conflicting objectives. The elitist nondominated sorting genetic algorithm [37] (called, i-NSGA-II) is an effective way to solve the multiple-objective problem. In this article, we propose a multiple-objective optimization algorithm based on i-NSGA-II, referred to as MO-NSGA, to solve the JRORS problem which is divided into problems (22) and (23). In multiple-objective optimization problem, there is a single best solution for each objective. However, this solution may not fare for all the other objectives. Before presenting the MO-NSGA algorithm, we give some important descriptions and definitions.

Definition 1 (Nondominated Solutions [38]): The nondominated solutions are the ones that form the set of most interesting solutions for multiple-objective optimization problems.

Definition 2 (Crowding Distance [38]): The average distance of two adjacent solutions along each objective direction on either side of one particular solution is called the crowding distance.

A chromosome structure that represents a set of potential solution is expressed as: $p_K = \{a_1, a_2, \ldots, a_q | b_1, b_2, \ldots, b_q, a_q \in \{0, 1\}, b_q \in (0, R_n)\}$. The population $\mathbf{P}$ is composed of K chromosomes, forming a pool of potential solution. It is well known that parent inheritance plays an important role in determining the quality of offspring. In the framework of the multiple-objective optimization MO-NSGA algorithm, an improved initial random population **IP** of the best chromosome using nondominated sorting and crowding-distance calculation is introduced. The optimal solution p_{opt} is evolved by the ranking selection, direction-based crossover, and adaptive mutation operations of the population $\mathbf{P}$.

Definition 3 (Ranking Selection): The chromosomes in the population are first sorted according to their fitness, and ranking selection discards $\lfloor \rho_s K \rfloor$ amounts of chromosomes that are ranked relatively lower while replicating the same amounts of chromosomes that are ranked higher. Note that $\rho_s \in (0, 0.5]$ is a proportional parameter [39].

Definition 4 (Direction-Based Crossover): After ranking selection, the ordered sequence of the population is obtained, $\mathbf{\Psi} = \{\mathbf{\Psi}_1, \mathbf{\Psi}_2, \ldots, \mathbf{\Psi}_K | \mathbf{\Psi}_1 \geq \mathbf{\Psi}_2 \geq \cdots \geq \mathbf{\Psi}_K\}$. We define the notation $L \geq Q$ as the fitness of chromosome L is better or equal to that of the chromosome Q. The population is divided into the strong group and the weak group, denoted as $\Psi^S = \{\Psi_1^S, \Psi_2^S, \ldots, \Psi_A^S\}$ and $\Psi^W = \{\Psi_1^W, \Psi_2^W, \ldots, \Psi_A^W\}$, respectively, where $A = K/2$. The directed-based crossover operation is performed by the following rule:

$$\mathbf{\Psi}_i^{W*} \leftarrow \mathbf{\Psi}_i^W + \theta_{c,i} \overrightarrow{\mathbf{d}}_i, \qquad i = 1, 2, \ldots, A \tag{24a}$$

$$\mathbf{\Psi}_i^{S*} \leftarrow \mathbf{\Psi}_i^S + \theta_{c,i} \overrightarrow{\mathbf{d}}_i, \qquad i = 1, 2, \ldots, A \tag{24b}$$

where $\theta_{c,i}$ is the step size given by

$$\theta_{c,i} = \frac{\left| f(\Psi_i^W) - f(\Psi_i^S) \right|}{\max\{f(\Psi)\} - \min\{f(\Psi)\}} \tag{25}$$

and $\overrightarrow{\mathbf{d}}_i = [d_{i,1} d_{i,2} \cdots d_{i,n}]$ is the crossover direction vector given by

$$d_{i,j} = \begin{cases} 0, & \text{if } r_j < 0.5 \\ \mathbf{\Psi}_{i,j}^W - \mathbf{\Psi}_{i,j}^S, & \text{if } r_j \geq 0.5 \end{cases} \tag{26}$$

for $i = 1, 2, \ldots, A$, and $j = 1, 2, \ldots, n$, where n is the number of genes in a chromosome and r_j is a random number selected from (0, 1].

Definition 5 (Adaptive Mutation): Assume $\mathbf{\Psi}_k$ is a selected chromosome to be mutated, and $\mathbf{\Psi}_k \in [\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}]$, we give the following adaptive mutation rule:

$$\mathbf{\Psi}_k^1 = \mathbf{\Psi}_k + h\big(N_{\text{gen}}, \boldsymbol{x}_{\max} - \mathbf{\Psi}_k\big) \tag{27a}$$

$$\mathbf{\Psi}_k^2 = \mathbf{\Psi}_k - h\big(N_{\text{gen}}, \mathbf{\Psi}_k - \boldsymbol{x}_{\min}\big) \tag{27b}$$

where $h(N_{\text{gen}}, y)$ is the function associated with the generation

$$h\big(N_{\text{gen}}, y\big) = y\left(1 - r_m^{(1 - N_{\text{gen}}/N_{\max})^{\eta}}\right) \tag{28}$$

Algorithm 2 MO-NSGA

1: **repeat**
2: **Initialize** the parent population $\mathbf{P} = \{p_1, p_2, \ldots, p_K\}$
3: **Evaluate fitness:** $\{f(p_i), i = 1, 2, \ldots, K\}$
4: Non-dominated sorting and crowding-distance calculation
5: **Select** the best chromosomes with rank = 1 into the improved parent population **IP**.
6: **Until** $N_{IP} = K$
7: $N_{gen} = 1$
8: **While** $N_{gen} \leq N_{max}$ **do**
9: Non-dominated sorting and crowding-distance calculation
10: **Ranking selection:** $\mathbf{\Psi}(N_{gen}) \rightarrow$ **IP′**
11: **Direction-based crossover:** $\{\mathbf{\Psi}^{W*}, \mathbf{\Psi}^{S*}\}$
12: **Adaptive mutation**: $\mathbf{\Psi}'(N_{gen}) \rightarrow$ **IP″**
13: **Combine** chromosomes in **IP′** and **IP″** **NIP**
14: **Select** the best chromosomes from **NIP** $\rightarrow$ **IP**
15: **if** the rank of all chromosomes in **IP** are the same **do**
16: **break**
17: **end if**
18: **end While**
19: **Output**: the optimal solution $\rightarrow p_{opt}$

TABLE II
AVERAGE NODE DEGREE VERSUS TIME

Parameter	Value
Number of mobile users U	{12,20,32,40,52,60,72,80,92,100}
Number of BSs N	{3,5,8,10,13,15,18,20,23,25}
Workload of request w_q	1000-2000 (MHz)
Input data of request I_q	600-1000 (KB)
Priority of request q pr_q	(0, 1)
Ideal delay of request q Tg_q	[0.4, 0.6] (s)
Tolerable delay of request q Tb_q	Tg_q + [0.1, 0.15] (s)
Computing capacity of BS n R_n	{60,70, 80} (GHz)
Computing capacity of macro-BS Rc	120 (GHz)
The fixed bandwidth B	20 (MHz)
The fixed altitude of BS H	10 (m)
Noise power σ_o^2	-100 (dBm)
The maximum transmitting power of mobile user	{4, 5, 6} (w)

where N_{gen} is the current generation number and $N_{\max}$ is the maximum number of generations, $r_m \in (0, 1]$ and $\eta \in [2, 5]$. To sum up, we give the specific pseudocode of MO-NSGA as shown in Algorithm 2.

Therefore, the PA problem can be addressed using the quasiconvex technique and solved by NCGG as shown in Algorithm 1. Besides, we analyze the JRORS problem as a double decision-making problem which can be solved by Algorithm 2.

V. PERFORMANCE EVALUATION

A. Simulation Settings

To evaluate the effectiveness of the proposed algorithms, we implemented the NCGG and MO-NSGA algorithms using MATLAB R-2019a. The simulations were conducted on an Intel i5, 3.7-GHz PC with 8-GB RAM. We consider a multizone edge computing system consisting of mobile users, multiple BSs, and a macro-BS. Each BS equipped with an edge server and covers a zone. We quantize a mobile user into a zone associated with the BS based on the location of the mobile user and the area covered by the BS. The parameters of the simulation are shown in Table II.

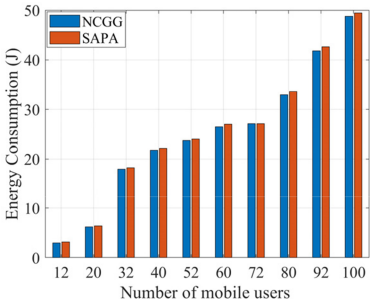

Fig. 3. Energy consumption versus number of mobile users.

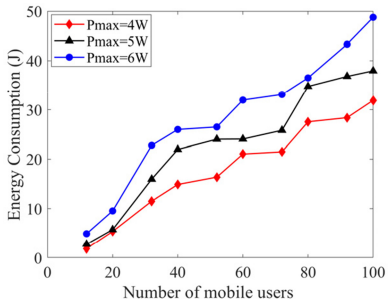

Fig. 4. Energy consumption versus $p_{\max}$ (NCGG).

B. Approaches

We compare the performance of the NCGG and MO-NSGA algorithms with the following approaches.

1) *Simulated Annealing PA (SAPA):* Simulated annealing (SA) as a classical heuristic algorithm can solve the NP-hard problem efficiently. We apply SA to solve the PA problem defined in Section IV-A.
2) *Yalmip:* Yalmip is a free optimization tool developed by Lofberg, which can solve the multiple-objective optimization problem. We use Yalmip to solve the JRORS problem defined in Section IV-B.
3) *Random RO and Greedy RS Strategy (ROGS) [40]:* Requests from the mobile users are randomly offloaded to the BSs to maximize the system welfare, and each BS schedules the computing resources greedily to maximize the system utility.
4) *Heuristic RO and Bisection-Based RS Strategy (HOBS) [10]:* A novel heuristic RO algorithm can find a local optimum in polynomial time, and RS can be solved by the bisection method.

C. Performance of NCGG

1) Energy Consumption Versus the Number of Mobile Nodes: In this case, the maximum power $p_{\max}$ of mobile users is set to 5 W. Fig. 3 shows the performance of the proposed NCGG compared to the SAPA. Obviously, as the number of mobile users increases, the total transmitting energy consumption increases. It is also observed that the energy consumption of NCGG is always smaller than that of SAPA under different number of mobile users, which implies that NCGG can give a better PA result compared to SAPA in energy saving. In NCGG, each mobile user gradually decreases from $p_{\max}$ to the subgradient direction, to find its optimal transmitting power and achieve the Nash equilibrium. However, in SAPA, the stable solution obtained in the process of random optimization may be a local optimal solution. If there is no special statement, the following experiments are based on the power policy P^* obtained by NCGG.

2) Energy Consumption Versus Maximum Power p_{max}: As shown in Fig. 4, under different maximum power $p_{\max}$, i.e., $p_{\max} = 4$, 5, and 6 W, we evaluate the energy consumption under different number of mobile users. It is noticed that the energy consumption is related to the maximum power $p_{\max}$, i.e., the bigger the $p_{\max}$ is, the higher the energy consumption is. This is because the average transmitting power of mobile users is higher under the bigger maximum power $p_{\max}$, which leads to more energy consumption.

3) Convergence Property of NCGG: To evaluate the convergence property of NCGG, the error of energy consumption E between adjacent iterations is set to 0.005, i.e., if the error of E is smaller than 0.005, the mobile user is considered to find its optimal solution. Fig. 5 shows the convergence property under different number of mobile users with $p_{\max} = 5$ W. From the convergence curve illustrated in Fig. 5(a), we know that the convergence rate is fast, and it only takes 64 iterations to converge when the number of mobile users is 40. Moreover, from Fig. 5(b), we know that it takes 123 iterations to converge when the number of mobile users is 100, which indicates that our proposed NCGG has a good convergence property. Table III summarizes the convergence iterations under different mobile users.

D. Performance of MO-NSGA

1) Effect of Number of Mobile Users: In this case, the computing capacity of all BSs are the same, i.e., $R_n = 70$ GHz, and all mobile users offload the same profile request with $w_q = 1500$ (Magacycles), $I_q = 700$ (KB), $Tg_q = 0.5$ (s), and $Tb_q = 0.65$ (s). We define the response rate as the ratio of the number of completed calculation to the total number of requests within the tolerant delay of the request. As shown in Fig. 6, we evaluate the performance including system welfare and response rate of MO-NSGA, compared to the other three algorithms against different number of mobile users. From Fig. 6(a), we observe that Yalmip as an optimization tool cannot achieve a good result. This is because the continuous relaxation in the JRORS problem is nonconvex, which means that the branching process of Yalmip is not guaranteed to find a globally optimal solution. Moreover, the MO-NSGA and HOBS perform equally well, and both perform better than ROGS and Yalmip. It can be seen that with the increasing number of mobile users, the system welfare increases. Fig. 6(b) illustrates that MO-NSGA outperforms the other compared approaches in response rate under different number of mobile users. It should be noted that MO-NSGA can achieve a high response rate even in the case of a large number of mobile users, which also reflects the extensibility of MO-NSGA.

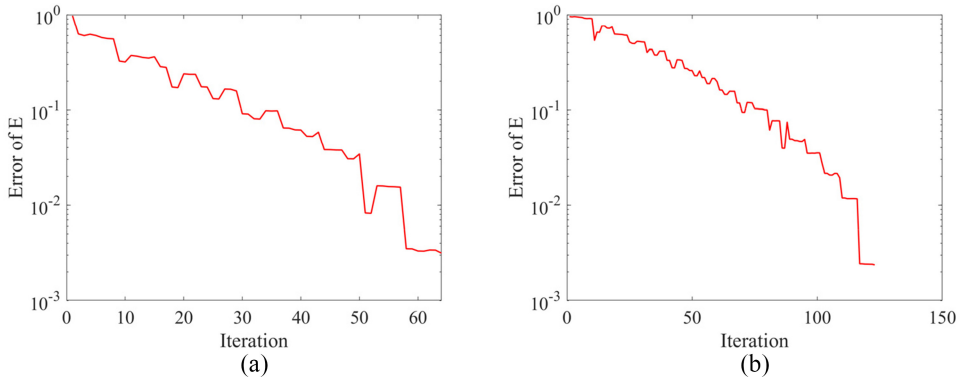

Fig. 5. Convergence property of NCGG versus different number of mobile users: (a) $U = 40$ and (b) $U = 100$.

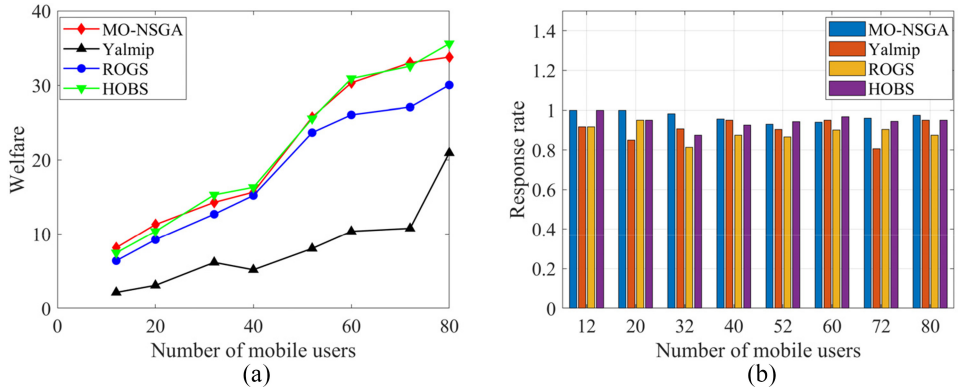

Fig. 6. Performance versus different number of mobile users. (a) Welfare. (b) Response rate.

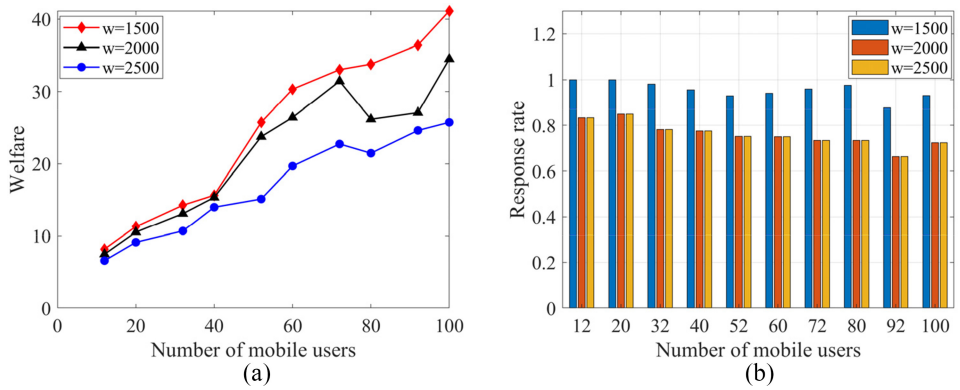

Fig. 7. Performance under different request workload. (a) Welfare. (b) Response rate.

TABLE III
AVERAGE NODE DEGREE OF NCGG VERSUS TIME

Number of mobile users U	**12**	**20**	**32**	**40**	**52**	**60**	**72**	**80**	**92**	**100**
Number of iterations	21	35	54	64	79	91	107	110	121	123

2) Effect of Request Workload: Here, we evaluate the performance of MO-NSGA under different request workload, $w_q = 1500$, 2000, and 2500. As shown in Fig. 7, we observe that with the decrease of request workload, both the system welfare and response rate increase. In particular, when the request workload exceeds 2000, the response rate decreases sharply. This is because, the computing resources of BS are not sufficient to be scheduled for offloading requests with more workloads, thus degrading the response rate and the system welfare. In the next experiments, the request workload is set to 1500 except for special cases.

3) Effect of Request Profile: In this case, different request profiles in terms of request workload w_q and request input size I_q are configured to evaluate the performance of MO-NSGA, compared with other approaches. The system welfare and response rate are plotted in Fig. 8(a) and (b) under

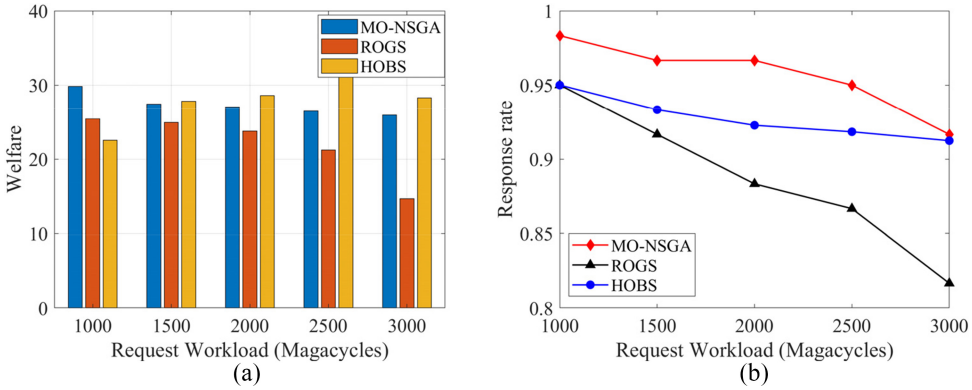

Fig. 8. Performance versus different request workload with $U = 60$ and $I_q = 700$ KB. (a) Welfare. (b) Response rate.

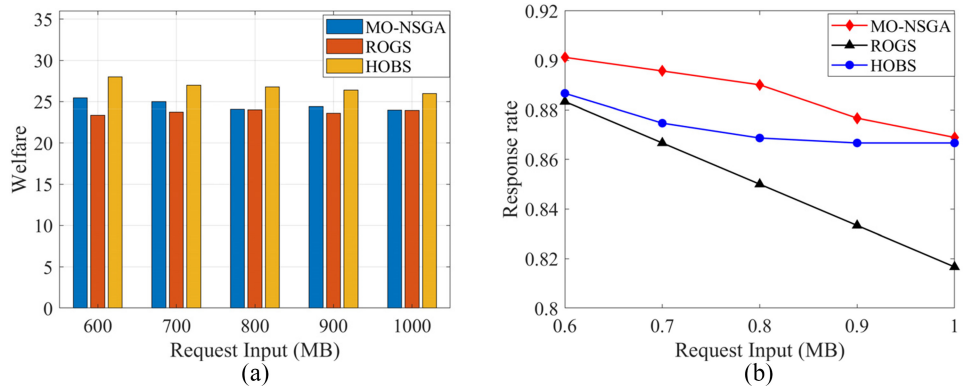

Fig. 9. Performance versus different request input with $U = 60$ and $w_q = 1500$ Magacycles. (a) Welfare. (b) Response rate.

different values of w_q, and we observe that MO-NSGA always outperforms ROGS and HOBS in response rate, but performs inferior to HOBS in system welfare. This implies that MO-NSGA sacrifices part of the system welfare to maximize the response rate in the optimization process. From Fig. 8(a), we observe that with the increase of w_q, the response rate decreases. It is evidently because the computing overhead of BS becomes higher as w_q increases, leading to more and more requests being unable to response in time. Similarly, as shown in Fig. 9(a) and (b), MO-NSGA always shows the best in response rate but performs inferior to HOBS in terms of system welfare. It can be seen that with the increase of l_q, the response rate also decreases, which is because a large amount of input data increases the transmitting delay. Therefore, we know that MO-NSGA shows the best performance in terms of response rate, even under different request profiles.

E. Heterogeneous Case

In this experiment, we consider a heterogeneous configuration of the system: $U = 48$, $N = 4$, R_n is randomly selected from $\{60, 70, 80\}$ GHz, w_q, I_q, Tg_q, and Tb_q of requests are generated from the values as shown in Table II. The BSs are fixed and mobile users are randomly placed in the area, as shown in Fig. 10, where the red triangle represents the BS and the blue dot represents the mobile user. We assume that the mobile users update their locations every time slot τ, $\tau = 1$ s.

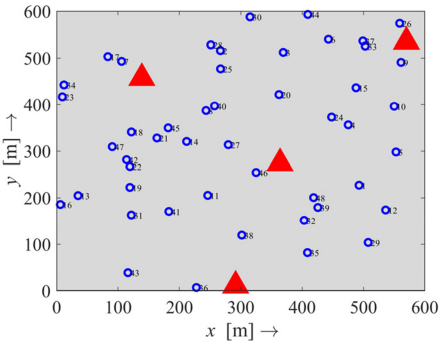

Fig. 10. Performance versus different request.

During each slot, every mobile user will generate one request. Hence, the PA policy P, RO policy X, and computing RS policy Y will change with the time, because the location of mobile users and the request profile is dynamic.

Fig. 11 shows the performance of MO-NSGA in response rate against the time slot. From the figure, we observe that in the most time slot the response rate can reach more than 0.85, and at some time slots, the response rate exceeds 0.95 or even reaches 1. However, the response rate is low than 0.8 at some time slots; this is because the profile of requests randomly generated by the mobile users may be too complex in these time slots. In general, MO-NSGA can maintain an average response rate of 0.8687 during the dynamic system.

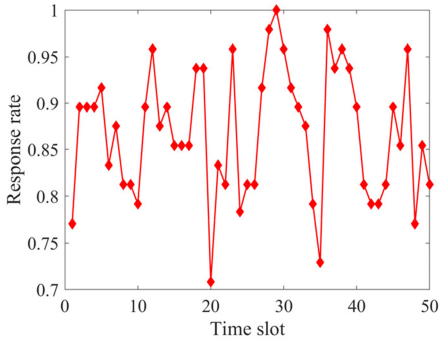

Fig. 11. Performance versus different request.

In summary, NCGG can give a better PA result in energy saving and has a good convergence property. Besides, MO-NSGA outperforms the other compared approaches in both welfare and response rate under different number of mobile users, different request workload, and request profile. Furthermore, MO-NSGA can also maintain an average response rate of 0.8687 during the dynamic system in a heterogeneous case.

VI. Conclusion

In this article, we studied the request scheduling problem in the UDEC network. We considered a UDEC network consisting of a macro-BS, many micro-BSs, and a large number of mobile users under the 5G architecture. The NOMA protocol was used as the multiple access scheme between users and BSs. In particular, we considered the interference between mobile users and BSs under the NOMA protocol, and we first formulated a PA problem and presented an NCGG to solve it. Then, we developed the problem involving jointly optimizing the RO for mobile users and the computing RS at the micro-BSs, by forming a mixed-integer nonlinear program. The problem was analyzed as a double decision-making problem, and we proposed a multiple-objective optimization algorithm based on i-NSGA-II (MO-NSGA) to address it. The simulation results verified our algorithms, in which NCGG can effectively save energy consumption, and MO-NSGA outperforms the existing approaches in terms of response rate and maintains a good performance in a dynamic MEC system. However, the proposed algorithm was not implemented in real-world applications. In the future, we will work on the design of edge computing RS algorithms for systems based on realistic applications, so as to solve the bottlenecks of practical problems.

Appendix

The first-order derivative of objective (16a) is calculated as follows:

$$\frac{\partial E_n(p_{un})}{\partial p_{un}} = \frac{I_u(h(p_{un})g(p_{un}) - \gamma p_{un})}{Bg(p_{un})h^2(p_{un})} \tag{29}$$

where

$$h(p_{un}) = \log_2(1 + \gamma p_{un}) \tag{30}$$

$$g(p_{un}) = \ln 2(1 + \gamma p_{un}) \tag{31}$$

and then we can obtain the second-order derivative of objective (16a)

$$\frac{\partial^2 E_n(p_{un})}{\partial p_{un}^2} = \frac{I_u\gamma(k(p_{un}) + \ln 2)}{Bg^2(p_{un})h^3(p_{un})} \tag{32}$$

where

$$k(p_{un}) = 2\gamma p_{un} - g(p_{un}). \tag{33}$$

To satisfy (29) being equal to 0, we get

$$\gamma p_{un}^* = h(p_{un}^*)g(p_{un}^*). \tag{34}$$

By substituting (34) into (32), we get

$$\frac{\partial^2 E_n(p_{un})}{\partial p_{un}^2} = \frac{I_u\gamma\left(\log_2(1 + \gamma p_{un}^*)^2/2 + \ln 2\right)}{Bg^2(p_{un}^*)h^3(p_{un}^*)}. \tag{35}$$

Hence, we can easily know that (35) is strictly positive $\forall p_{un}^* \in (0, P_{\max}]$. According to the conclusion in [41], the condition for satisfying a strict quasiconvex function is such that a variable satisfying first-order derivative of the function is equal to 0 and also satisfying the second-order derivative of the function is greater than 0. Therefore, objective (16a) is a strict quasiconvex function in $(0, P_{\max}]$.

References

[1] M. Smith, A. Maiti, A. D. Maxwell, and A. A. Kist, "Object detection resource usage within a remote real-time video stream," in *Online Engineering & Internet of Things*, Cham, Switzerland: Springer, 2018, pp. 266–277.

[2] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-time detection and tracking for augmented reality on mobile phones," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 3, pp. 355–368, May/Jun. 2010.

[3] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tut.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[5] "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022," San Jose, CA, USA, Cisco, White Paper, 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11–738429.html

[6] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, "Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 668–682, Mar. 2019.

[7] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[8] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resources optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[9] Q. Wang, S. Guo, J. Liu, and Y. Yang, "Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing," *Sustain. Comput. Informat. Syst.*, vol. 21, pp. 154–164, Mar. 2019.

[10] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[11] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[12] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Trans. Cloud Comput.*, vol. 7, no. 1, pp. 196–209, Jan.–Mar. 2019.

[13] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 72–79, Feb. 2016.
[14] J. Wu, Z. Zhang, H. Yu, and Y. Wen, "Cloud radio access network (C-RAN): A primer," *IEEE Netw.*, vol. 29, no. 1, pp. 35–41, Jan./Feb. 2015.
[15] L. Dai, B. Wang, Y. Yuan, S. Han, C.-L. I, and Z. Wang, "Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 74–81, Sep. 2015.
[16] Z. Ding, M. Peng, and H. V. Poor, "Cooperative non-orthogonal multiple access in 5G systems," *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1462–1465, Aug. 2015.
[17] S. M. R. Islam, N. Avazov, O. A. Dobre, and K.-S. Kwak, "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 721–742, 2nd Quart., 2017.
[18] S. Chen, F. Qin, B. Hu, X. Li, and Z. Chen, "User-centric ultra-dense networks for 5G: Challenges, methodologies, and directions," *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 78–85, Apr. 2016.
[19] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2522–2545, 4th Quart., 2016.
[20] D. López-Pérez, M. Ding, H. Claussen, and A. H. Jafari, "Towards 1 Gbps/UE in cellular systems: Understanding ultra-dense small cell deployments," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2078–2101, 4th Quart., 2015.
[21] A. Osseiran *et al.*, "Scenarios for 5G mobile and wireless communications: The vision of the METIS project," *IEEE Communi. Mag.*, vol. 52, no. 5, pp. 26–35, May 2014.
[22] B. Yu, L. Pu, Q. Xie, and J. Xu, "Energy efficient scheduling for IoT applications with offloading, user association and BS sleeping in ultra dense networks," in *Proc. 16th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, Shanghai, China, 2018, pp. 1–6.
[23] C. Ma, F. Liu, Z. Zeng, and S. Zhao, "An energy-efficient user association scheme based on robust optimization in ultra-dense networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC Workshops)*, Beijing, China, 2018, pp. 222–226.
[24] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
[25] T. V. Do, N. H. Do, H. T. Nguyen, C. Rotter, A. Hegyi, and P. Hegyi, "Comparison of scheduling algorithms for multiple mobile computing edge clouds," *Simulat. Model. Pract. Theory*, vol. 93, pp. 104–118, May 2019.
[26] L. Gu, J. Cai, D. Zeng, Y. Zhang, H. Jin, and W. Dai, "Energy efficient task allocation and energy scheduling in green energy powered edge computing," *Future Gener. Comput. Syst.*, vol. 95, pp. 89–99, Jun. 2019.
[27] Y. Jie, X. Tang, K.-K. R. Choo, S. Su, M. Li, and C. Guo, "Online task scheduling for edge computing based on repeated Stackelberg game," *J. Parallel Distrib. Comput.*, vol. 122, pp. 159–172, Dec. 2018.
[28] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2082–2091, Dec. 2017.
[29] K. Lin, S. Pankaj, and D. Wang, "Task offloading and resource allocation for edge-of-things computing on smart healthcare systems," *Comput. Elect. Eng.*, vol. 72, pp. 348–360, Nov. 2018.
[30] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, "A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4831–4843, Jun. 2019.
[31] T. Bahreini, H. Badri, and D. Grosu, "An envy-free auction mechanism for resource allocation in edge computing systems," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Seattle, WA, USA, 2018, pp. 313–322.
[32] S. Misra and N. Saha, "Detour: Dynamic task offloading in software-defined fog for IoT applications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1159–1166, May 2019.
[33] H. Guo, J. Liu, and J. Zhang, "Computation offloading for multi-access mobile edge computing in ultra-dense networks," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 14–19, Aug. 2018.
[34] H. Guo, J. Zhang, J. Liu, H. Zhang, and W. Sun, "Energy-efficient task offloading and transmit power allocation for ultra-dense edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.
[35] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.
[36] Y. Nakamura, T. Mizumoto, H. Suwa, Y. Arakawa, H. Yamaguchi, and K. Yasumoto, "In-situ resource provisioning with adaptive scale-out for regional IoT services," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Seattle, WA, USA, 2018, pp. 203–213.
[37] M. Kumar and C. Guria, "The elitist non-dominated sorting genetic algorithm with inheritance (i-NSGA-II) and its jumping gene adaptations for multi-objective optimization," *Inf. Sci.*, vols. 382–383, pp. 15–37, Mar. 2017.
[38] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 2000, pp. 849–858.
[39] Y.-C. Chuang, C.-T. Chen, and C. Hwang, "A real-coded genetic algorithm with a direction-based crossover operator," *Inf. Sci.*, vol. 305, pp. 320–348, Jun. 2015.
[40] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.
[41] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Math. Oper. Res.*, vol. 23, no. 4, pp. 769–805, 1998.

Shihong Hu received the bachelor's degree in communication engineering from Jiangnan University, Wuxi, China, in 2016, and the Ph.D. degree from the School of Internet of Things Engineering, Jiangnan University.

Her research interests include sensor networks and edge computing.

Guanghui Li received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005.

He is currently a Professor with the Department of Computer Science, Jiangnan University, Wuxi, China. His research was supported by the National Foundation of China, Zhejiang, Jiangsu Provincial Science and Technology Foundation, and other Governmental and Industrial Agencies. He has published over 70 papers in journal or conferences. His research interests include wireless sensor networks, fault tolerant computing, and nondestructive testing and evaluation.