When Mobile-Edge Computing (MEC) Meets Nonorthogonal Multiple Access (NOMA) for the Internet of Things (IoT): System Design and Optimization

Jianbo Du, *Member, IEEE*, Wenhuan Liu, Guangyue Lu, Jing Jiang, *Member, IEEE*, Daosen Zhai, *Member, IEEE*, F. Richard Yu, *Fellow, IEEE*, and Zhiguo Ding, *Fellow, IEEE*

***Abstract*—Mobile-edge computing (MEC) is considered as a promising technology to enable low latency applications while consuming less energy, and nonorthogonal multiple access (NOMA) is regarded as a hopeful method of increasing spectrum efficiency and the wireless network capacity. In this article, we consider a NOMA-MEC-based Internet-of-Things (IoT) network, and propose a joint optimization framework to maximize the effective system capacity, i.e., the number of IoT devices whose tasks are processed successfully, and meanwhile to maximize the total energy saving. First, we concentrate on improving the effective system capacity from the wireless side by introducing NOMA, and from the IoT device side by task offloading decision optimization, where distributed optimization is conducted and closed-form solution is obtained. Then, we maximize the total energy saving also from two aspects, i.e., the device-side computation resource allocation, and the wireless side joint admission control, user clustering, orthogonal subcarrier assignment, and transmit power control, where we resort to graph theory and propose a low-complexity heuristic algorithm to solve it. Abundant simulation results demonstrate our proposed joint optimization algorithm performs well in both effective system capacity optimization and energy saving maximization.**

***Index Terms*—Admission control, computation offloading, nonorthogonal multiple access (NOMA), resource allocation, user clustering.**

I. Introduction

WITH the rapid development of the Internet of Things (IoT), our daily life and communication mode have witnessed great changes in recent years [1]. Large amounts of intelligent user equipments and other smart IoT devices are flooding into wireless networks, meanwhile various smart applications have been flourishing, such as smart city, smart grid, smart manufacturing, smart health monitoring, automatic driving [2], etc.

While IoT brings great convenience to our daily life, it also brings some tricky challenges.

1) Most smart applications are usually latency critical, requiring strong processing capacities, while IoT devices are generally small in size and difficult to afford the processing of those sophisticated smart applications [3]–[5].
2) Smart applications are generally computational intensive and energy demanding, large quantities of energy will be consumed in the task executing process. However, IoT devices are usually resource constrained and nonrechargeable, and the standby time will be reduced greatly in running those tasks directly.
3) Because of the extreme scarcity of wireless resources, today's networks are increasingly incompetent to provide massive connectivity and satisfactory wireless communications [6].

To address the first two challenges, mobile-edge computing (MEC) [7], [8] has been regarded as a promising technology to enhance the processing performance of smart tasks, and economize energy for IoT devices. The core concept of MEC is to integrate powerful MEC servers into radio access networks, e.g., WiFi access points or base stations, etc., in close proximity to IoT devices. Through offloading computational-intensive and energy-demanding tasks to adjacent MEC servers [9], [10], low-latency high-reliable user services can be enabled at IoT

Manuscript received September 18, 2020; accepted November 14, 2020. Date of publication December 30, 2020; date of current version May 7, 2021. This work was supported in part by the Natural Science Foundation of China under Grant 61901367; in part by the Natural Science Foundation of Shaanxi Province under Grant 2020JQ-844; in part by the Science and Technology Innovation Team of Shaanxi Province for Broadband Wireless and Application under Grant 2017KCT-30-02; in part by the National Science and Technology Major Project under Grant 2016ZX03001016; in part by the National Natural Science Foundation of China under Grant 61871321, Grant 61901381, Grant 62001357, and Grant 62071377; in part by the National Key Research and Development Program of China under Grant 2018YFE0126000; in part by the Key Research and Development Plan of Shaanxi Province under Grant 2017ZDCXL-GY-05-01; and in part by the Xi'an Key Laboratory of Mobile Edge Computing and Security under Grant 201805052-ZD3CG36. *(Corresponding authors: Guangyue Lu; Jing Jiang.)*

Jianbo Du, Wenhuan Liu, Guangyue Lu, and Jing Jiang are with the Shaanxi Key Laboratory of Information Communication Network and Security, School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China (e-mail: dujianboo@163.com; 1610613786@qq.com; tonylugy@163.com; jiangjing@xupt.edu.cn).

Daosen Zhai is with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: zhaidaosen@nwpu.edu.cn).

F. Richard Yu is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: richard.yu@carleton.ca).

Zhiguo Ding is with the School of Electrical and Electronic Engineering, University of Manchester, Manchester M13 9PL, U.K. (e-mail: zhiguo.ding@manchester.ac.uk).

Digital Object Identifier 10.1109/JIOT.2020.3041598

devices, and/or more energy can be economized [11], and therefore, the battery life of IoT devices can be prolonged [12]. Apart from the advantages, MEC also brings about extra wireless data transmission between IoT devices and MEC servers in task offloading, which however, may neutralize the advantage of MEC in the scenarios with large amounts of user equipments and limited radio resources [13].

Nonorthogonal multiple access (NOMA) [14], [15] has been considered as a potential and compelling solution to improve the spectrum efficiency, and thus to increase the wireless transmit rate and the number of accommodated users. Therefore, it could be employed to overcome the above third challenge in wireless data transmission in IoT systems [16]. NOMA utilizes the diversity in the power domain or code domain, and permits multiple users to share the same radio resource simultaneously. By implementing successive interference cancelation (SIC) at user side, the overlapping signals of different users can be decoded successfully [17], [18]. Therefore, NOMA performs good in improving both spectrum efficiency and system capacity, and thus caters for the demands of IoT systems on massive connections, higher data rates, ultra low transmission latencies, and also good in increasing the data transmission performance in the task offloading process of MEC systems [13], [19].

Motivated by their respective advantages, it may be beneficial to combine NOMA with MEC for better task offloading performance [20], [21]. However, several challenging issues should first be considered and addressed. First, offloading too many tasks to MEC server may lead to severe cochannel interference in the NOMA data transmission process, which will not only increase the complexity and economical cost of IoT devices but also will increase the transmission delay and may lead to frustrated task processing performance. Therefore, brilliant offloading decision-making strategies are necessary. Second, in order to provide as many IoT devices with successful task offloading and meanwhile with preferable task processing performance, we should judiciously determine which IoT devices should be admitted in the wireless access network, which IoT devices should be clustered together into a group, and which subcarriers should be allocated to which clusters, etc. Third, taking into account the limited energy budget of IoT devices, we need to properly budget how much computation resource should be allocated for local processing, and how much transmit power should be assigned for task offloading, etc.

These considerations motivate the study of this article and the main contributions of our work are summarized as follows.

1) We investigate the *effective system capacity* (i.e., the number of IoT devices whose tasks are processed successfully within the maximum tolerable latency requirement) and the total saved energy maximization issues in a NOMA-MEC-enabled IoT network, by a joint optimization of users' offloading decision making, local CPU frequency allocation, access control, user clustering, subcarrier assignment, as well as transmit power control, with the maximum tolerable task processing latency guaranteed.
2) In order to improve the effective system capacity, we, on one hand, use NOMA in the wireless access network to accommodate as many IoT devices for task offloading , and on the other hand, perform offloading decision optimization to encourage as many local-feasible IoT devices to process their tasks locally, thus to spare as many subcarriers for the IoT devices when local processing is infeasible.
3) To maximize the total economized energy of all IoT devices, we first optimize computation resource allocation on the device side, where the closed-form solution is obtained; moreover, we perform admission control, user clustering, subcarrier assignment, and transmit power control on the MEC server side, where by resorting to the concept of maximum weighted independent set (MWIS) in the graph theory, we propose a low-complexity heuristic algorithm and suboptimal solutions can be obtained.
4) Extensive simulation results demonstrate our proposed algorithms perform well in terms of both effective system capacity maximization and energy saving optimization.

The remainder of this article is organized as follows. Related works are presented in Section II. Section III introduces the system model and problem formulation. In Sections IV and V, the problem is solved efficiently. The simulation results are provided in Section VI. Finally, this article is concluded in Section VII.

II. Related Works

The study of MEC and NOMA have attracted extensive studies in their own area, and in recent years, the combination of the both has become a new hot research topic, thanks to their own advantages. Many approaches have been presented mainly from the following aspects in multiuser NOMA-based MEC systems.

Owing to the limited energy supply of IoT devices, many studies concentrated themselves on energy consumption minimization issues in NOMA-based MEC systems. By jointly optimizing the transmit time and power allocation, Ding *et al.* [19] studied the energy minimization for a NOMA-based MEC system, where the problem was transformed into a geometric programming problem and the closed-form solution was obtained. Pan *et al.* [22] considered a MEC system using NOMA for both uplink task offloading and downlink results returning. By jointly optimizing the transmit power control, time allocation, and task offloading partitions, the total energy consumption was minimized and an iterative algorithm was developed employing successive convex approximation. Wang *et al.* [23] studied the latency-constrained weighted total system energy minimization in NOMA-based MEC systems, by jointly optimizing the offloading decision making, radio resource allocation, and the BS's SIC decoding order under partial and binary offloading scenarios, respectively, where the Lagrange dual decomposition, branch-and-bound, greedy method and convex relaxation were used for problem solving. Song *et al.* [24] proposed a partial task-offloading scheme in a NOMA-based MEC heterogeneous network, where radio and computation resource allocation were jointly optimized to minimize the energy consumption of all users under the tolerable latency constraint. Zeng and Fodor [25] studied joint communication and computation resource allocation in a NOMA-MEC system, assuming all the users offload their tasks to the MEC server, and a heuristic algorithm-based solution was proposed for energy minimization. Li *et al.* [26] investigated the total transmit power minimization in a wireless IoT network, and a genetic algorithm-based strategy was proposed to obtain the suboptimal solution to task-offloading decision making, subcarrier allocation, and transmit power control.

Many other works put their emphasis on latency reduction to enable delay-sensitive applications. Ding *et al.* [17] proposed to minimize the delay of task offloading in a NOMA enabled MEC system, where the formulated problem was transformed into a fractional programming problem and solved by two proposed iterative algorithms. Different form [17], Qian *et al.* [27] intended to minimize the maximum task execution latency in a MEC-NOMA-aware NB-IoT system. Employing convex optimization and heuristic algorithms, the optimal joint SIC ordering and computation resource allocation was obtained. In order to improve the efficiency of wireless data transmission and therefore to minimize the total delay in task offloading, Wu *et al.* [28] exploited NOMA-enabled multiaccess MEC, by jointly optimizing the users' offloaded workloads and the NOMA transmission-time. Wu *et al.* [13] proposed a NOMA-enabled partial computation offloading scheme, where NOMA was employed in both uplink task offloading and downlink result returning, in order to minimize the overall delay of completing all users' tasks. The optimization was achieved by jointly optimizing the portion of offloaded workloads, and the transmit duration in uplink task offloading and downlink result returning.

Some other works focused on both energy and delay reduction. Liu *et al.* [29] formulated a joint offloading decision making, user scheduling, and resource allocation problem in a NOMA-enabled multiuser mixed fog/cloud computing system, in order to minimize the weighted sum of total delay and energy consumption of all the users, and the problem was solved using alternating direction method of multipliers in an efficient distributed manner. Ding *et al.* [30] systematically analysed the performance of using NOMA in MEC for both uplink and downlink transmissions, where analytical results show that both the task processing latency and energy consumption can be reduced effectively by introducing NOMA into MEC systems. In order to minimize the weighted sum of the energy consumption and delay of all users, Diao *et al.* [31] jointly optimized the computation resource, power, and subcarrier allocations, where particle swarm optimization, game theory, and iterative algorithm was utilized and suboptimal solutions are obtained.

Except for energy and latency minimization, there are also some other research topics. Wei and Jiang [32] proposed an optimal computation offloading scheme with downlink NOMA. To achieve the maximal system utility, which is defined as the amount of data processed by fog nodes, the transmit power and the input data size delivered to the IoT device's task buffer are jointly optimized. Wen *et al.* [33] intended to maximize the energy efficiency in downlink NOMA fog networks by joint subchannel allocation and transmit power control optimization.

The above [17]–[33] have done a lot of works upon using NOMA in the task processing and/or results returning process in MEC systems. Some put their concentrate on spectrum efficiency improvement in task offloading, and therefore to obtain higher transmit rate, lower latency, and/or less energy consumption in task offloading. Some cared about increasing the number of served user equipments from the server side optimization by employing NOMA technology. In the following, we will perform optimization form both user and server sides, in order to economize more energy consumption and meanwhile to increase the effective system capacity as much as possible.

III. System Model and Problem Formulation

In this section, we will start with a detailed introduction to the concerned scenario, and then give our problem formulation.

A. Description of the Concerned Scenario

We consider a NOMA-MEC-enabled IoT system, which consists of N IoT devices, one MEC server, and K orthogonal subcarriers. The sets of IoT devices and subcarriers are denoted as $\mathcal{N} = \{1, 2, \ldots, N\}$ and $\mathcal{K} = \{1, 2, \ldots, K\}$, respectively, where the number of IoT devices N far outweighs that of subcarriers K, i.e., $N >> K$. The MEC server possesses stronger task processing capabilities, while the IoT devices have certain and may not be sufficient computation resources.

Each IoT device n has only one inseparable computation task, which can be represented by $J_n = \{D_n, \lambda_n, T_n^{\max}\}$, $n \in \mathcal{N}$, where D_n is the size of input data (in bits), λ_n is processing density (in CPU cycles/bit), indicating the complexity of task J_n's programme, and $T_n^{\max}$ is the maximum tolerable latency (in second) within which the task should be executed successfully [34]. The programme is backed up in the MEC server, while the input data, with a size of D_n, should be transferred to the MEC server in task offloading [35], [36]. Let x_n denote the offloading decision of each IoT device n, where $x_n = 0$ indicates the task is executed locally, and $x_n = 1$ otherwise.

1) Local Processing: Let f_n^{loc} and p_n^{loc} be the local task processing capability (in CPU cycles/s) and power consumption (in watt) of IoT device n, respectively, where f_n^{loc} takes its value among $[0, f_n^{\max}]$. If task J_n is implemented locally, the power consumption of IoT device n can be given by $p_n^{\text{loc}} = \alpha(f_n^{\text{loc}})^3$, where α is a constant coefficient about the CPU chip architecture [5]. Then, the task processing delay and energy consumption in local processing mode can be give by $T_n^{\text{loc}} = D_n\lambda_n/f_n^{\text{loc}}$ and $E_n^{\text{loc}} = \alpha D_n\lambda_n(f_n^{\text{loc}})^2$, respectively.

2) Task Offloading: If task J_n is going to be processed at the MEC server, IoT device n needs to transfer the input data to the MEC server through shared orthogonal subcarriers. For analytical tractability, we suppose the MEC server will start to process task J_n only when it has successfully received all the input data form the IoT device. The MEC server possesses stronger task processing capabilities and can allocate each IoT device with sufficient computation resources, nevertheless, the wireless resources will become the main bottleneck in task offloading.

In order to support huge amounts of IoT devices, traditional orthogonal multiple access (OMA) finds great difficulties owning to its exclusiveness and low-spectral efficiency. Meanwhile, since the input data sizes of IoT tasks are usually very small, it is rather wasteful to use OMA in IoT scenarios. Considering the above aspects, we assume each IoT device can be allocated with only one subcarrier, and all the IoT devices performing task offloading share K subcarriers using NOMA for data transmission in task offloading. To alleviate co-channel interference brought by NOMA, the MEC server utilizes the SIC technique to decode the multiple overlapping signals sequentially, where the optimal decoding order is the descending order in terms of channel power gain. Considering the tolerable actual SIC decoding complexity, we suppose each subcarrier can only serve at most two IoT devices, in order to make a balance between performance degradation and effective system capacity improvement.

Let $\mathbf{S} = \{s_{n,k}\}$ be the subcarrier allocation matrix, where element $s_{n,k} = 1$ represents subcarrier k is allocated to IoT device n, and $s_{n,k} = 0$ otherwise. Let $\mathbf{p} = \{p_1, \ldots, p_N\}$ indicate the transmit power control vector, where p_n is the transmit power of IoT device n. Denote the channel gain on subcarrier k used by IoT device n as $g_{n,k}$, which includes path loss, shadowing, and fading. Given radio resource allocation $\mathbf{S}$ and $\mathbf{p}$, the signal-to-interference-plus-noise ratio (SINR) of IoT device n on subcarrier k can be given by

$$\gamma_{n,k} = \frac{x_n s_{n,k} p_n g_{n,k}}{\sum_{\substack{i\in\mathcal{N}\\ g_{i,k}<g_{n,k}}} x_i s_{i,k} p_i g_{i,k} + \sigma^2} \tag{1}$$

where σ^2 is the noise power. Then, the transmit rate of IoT device n on subcarrier k can be given by $R_n^k = B_0 \log_2(1 + \gamma_{n,k})$, and consequently, the transmit rate of IoT device n can be given by

$$R_n = \sum_{k\in\mathcal{K}} B_0 \log_2\left(1 + \gamma_{n,k}\right). \tag{2}$$

Suppose the MEC server has stronger processing capability, and can allocate each IoT device with sufficient computation resource f^{mec} for task processing, then the delay and energy consumption of IoT device n in the task-offloading model can be given by

$$T_n^{\text{mec}} = \frac{D_n}{R_n} + \frac{D_n\lambda_n}{f^{\text{mec}}} \tag{3}$$

$$E_n^{\text{mec}} = \frac{D_n}{R_n} p_n + \frac{D_n\lambda_n}{f^{\text{mec}}} p_n^{\text{id}} \tag{4}$$

where p_n^{id} is the idle power of IoT device n.

3) Admission Control: Since the number of orthogonal bandwidth subcarriers is much less than that of IoT devices, and each subcarrier can accommodate at most two IoT devices, then, at most $2K$ IoT devices can be admitted in for task offloading, and the rest local-infeasible and not-admitted-in IoT devices will fail in task processing. Let y_n be the admission control variable, where $y_n = 1$ denotes IoT device n is permitted to access the wireless network, and $y_n = 0$ otherwise.

4) Energy Saving: Considering offloading decision and admission control, the delay and energy consumption of each IoT device n is given by

$$T_n = (1 - x_n) T_n^{\text{loc}} + x_n y_n T_n^{\text{mec}} \tag{5}$$

$$E_n = (1 - x_n) E_n^{\text{loc}} + x_n y_n E_n^{\text{mec}}. \tag{6}$$

Compared with local processing where no optimization is performed, the saved energy of IoT device n can be given by

$$\begin{aligned} E_n^{\text{save}} &= \alpha D_n \lambda_n \left(f_n^{\max}\right)^2 - E_n \\ &= \alpha D_n \lambda_n \left(f_n^{\max}\right)^2 - (1 - x_n) E_n^{\text{loc}} - x_n y_n E_n^{\text{mec}}. \end{aligned} \tag{7}$$

B. Problem Formulation

We propose to maximize the total economized energy as well as the effective system capacity, and formulate the problem as the joint optimization of offloading decision $\mathbf{x} = \{x_1, \ldots, x_N\}$, admission control $\mathbf{y} = \{y_1, \ldots, y_N\}$, subcarrier assignment and user clustering $\mathbf{S} = \{s_{n,k}\}$, local computational resource allocation $\mathbf{f}^{\text{loc}} = \{f_1^{\text{loc}}, \ldots, f_N^{\text{loc}}\}$, and transmit power control $\mathbf{p} = \{p_1, \ldots, p_N\}$ as follows:

$$\begin{aligned} (\mathcal{P}_1):\ & \max_{\mathbf{x},\mathbf{f}^{\text{loc}},\mathbf{y},\mathbf{p},\mathbf{S}} \sum_{n\in\mathcal{N}} E_n^{\text{save}} \\ \text{s.t.}\ & (\text{C1}): T_n \le T_n^{\max} \quad \forall n \in \mathcal{N} \\ & (\text{C2}): 0 \le p_n \le p_n^{\max} \quad \forall n \in \mathcal{N} \\ & (\text{C3}): \sum_{k\in\mathcal{K}} s_{n,k} = 1 \quad \forall n \in \mathcal{N} \\ & (\text{C4}): \sum_{n\in\mathcal{N}} s_{n,k} \le 2 \quad \forall k \in \mathcal{K} \\ & (\text{C5}): s_{n,k} \in \{0, 1\} \quad \forall n \in \mathcal{N} \quad \forall k \in \mathcal{K} \\ & (\text{C6}): x_n \in \{0, 1\} \quad \forall n \in \mathcal{N} \\ & (\text{C7}): y_n \in \{0, 1\} \quad \forall n \in \mathcal{N} \\ & (\text{C8}): 0 \le f_n^{\text{loc}} \le f_n^{\max} \quad \forall n \in \mathcal{N} \end{aligned} \tag{8}$$

where (C1) means each task should be accomplished within a tolerable deadline; (C2) is the constraint on transmit power control, (C3) indicates that each IoT device is allocated with only one subcarrier; (C4) and (C5) together guarantee that each subcarrier could support at most two IoT devices; (C6) and (C7) are the binary constraints on offloading decision and admission control optimization; and (C8) is the constraint on local computation resource allocation.

A summary of the mainly used notations are presented in Table I.

IV. Problem Solving

In this section, we will first analyze the structure of our formulated joint optimization problem ($\mathcal{P}_1$), and then propose low-complexity algorithms to solve it.

A. Problem Structure Analyzing

It is generally quite troublesome to solve problem ($\mathcal{P}_1$) as a result of its mixed combinatory characteristics and the coupling in both the objective and the constraints [37], [38]. Therefore, it is necessary to propose easy-to-implement low-complexity algorithms, which is especially meaningful in

TABLE I
NOTATION DEFINITIONS

Symbol	Definition
D_n	The input data size of the task of IoT device n (in bits)
λ_n	Processing density of the task of IoT device n (in CPU cycles)
f_n^{max}	Maximum local processing capability of each IoT device (in CPU cycles/s)
T_n^{max}	Maximum tolerable latency of each task (in second)
f^{mec}	The computation resource allocated to each offloaded task (in CPU cycles/s)
$g_{n,k}$	The channel gain of IoT device n on subcarrier k
$\mathbf{x}$, x_n	Offloading decision vector and offloading decision of IoT device n
$\mathbf{f}^{loc}$, f_n^{loc}	Local CPU clock frequency vector of all IoT devices and of IoT device n
$\mathbf{S}$	Matrix of all IoT devices' subcarrier allocation
s_{nk}	Indicator of whether subcarrier k is allocated to IoT device n
$\mathbf{p}$, p_n	Transmit power control vector and transmit power of IoT device n
$\mathbf{y}$, y_n	Admission control vector
$\mathcal{N}$,N	The set and number of all IoT devices
$\mathcal{N}_1$,N_1	The set and number of all non-local-processing IoT devices
$\mathcal{N}_{off}$,N_{off}	The set and number of all task offloading IoT devices
$\mathcal{N}_{fail}$,N_{fail}	The set and number of all IoT devices whose task will not be processed
α	Coefficient used to model local processing energy consumption depending on chip architecture

large-scale situations of $(\mathcal{P}_1)$. For this purpose, we decouple $(\mathcal{P}_1)$ into two subproblems, one is local-related optimization subproblem, where offloading decision and local resource allocation are optimized distributedly at each IoT device; the other is the MEC server-side optimization subproblem, where admission control, user clustering and radio resource allocation are jointly optimized at MEC server. Next, we will solve the subproblems one after another. However, it should be noted that the proposed algorithms are possibly suboptimal after this decoupling [39], [40].

Lemma 1: The effective system capacity, i.e., the number of IoT devices whose tasks are processed successfully is maximized, on one hand, by encouraging as many IoT devices to perform their tasks locally when local processing is feasible, and on the other hand, by employing NOMA to accommodate as many IoT devices for task offloading [41].

Remark 1: Lemma 1 holds from the fact in two aspects. First, when the number of orthogonal subcarriers is far from sufficient to support large volume of IoT devices for successful task offloading, many offloaded tasks may fail in task processing when no subcarrier is allocated with, and consequently, the effective system capacity will shrink. From this point, when local processing is feasible, i.e., tasks can be accomplished with satisfactory QoS, we encourage the IoT devices to process their tasks locally, in order to spare as many subcarriers for those local-infeasible IoT devices. Thus, the effective system capacity is maximized on one hand. Second, when introducing NOMA into the wireless access network, each subcarrier can support more IoT devices compared with the traditional OMA technique, and the effective system capacity can be increased on the other hand.

Remark 2: When we encourage as many local-feasible IoT devices to process their tasks locally, the effective capacity is maximized from the device-side optimization, however, the economized energy may not be maximized, which is actual the normality. Nevertheless, we can take some remedial measures, i.e., local computation resource allocation, to maximize the saved energy form the IoT device side. Thus, not only the effective system capacity is maximized, but also the saved energy is increased as a result of device-side optimization.

B. Distributed Local-Related Optimization

Each IoT device n first assumes its task is processed locally, i.e., the offloading decision $x_n = 0$, and then determines the feasibility of the following problem $(\mathcal{P}_2)$, from which the offloading decision and local resource allocation strategies are obtained:

$$\begin{aligned}(\mathcal{P}_2):\ \max_{\mathbf{f}^{\text{loc}}}\ & \sum_{n\in\mathcal{N}}\left[\alpha D_n\lambda_n\left(f_n^{\max}\right)^2-\alpha D_n\lambda_n\left(f_n^{\text{loc}}\right)^2\right]\\ \text{s.t.}\quad & (\text{C1}):\ \frac{D_n\lambda_n}{f_n^{\text{loc}}}\le T_n^{\max}\quad \forall n\in\mathcal{N}\\ & (\text{C8}):\ 0\le f_n^{\text{loc}}\le f_n^{\max}\quad \forall n\in\mathcal{N}.\end{aligned}\tag{9}$$

In problem $(\mathcal{P}_2)$, the local computation resource allocation is independent in the objective and constraints, so the optimization of f_n^{loc} in each IoT device is independent form each other, and thus problem $(\mathcal{P}_2)$ can be decoupled into the optimization for each IoT device n as follows:

$$\begin{aligned}(\mathcal{P}_3):\ \min_{f_n^{\text{loc}}}\ & \alpha D_n\lambda_n\left(f_n^{\text{loc}}\right)^2\\ \text{s.t.}\quad & (\text{C1}):\ \frac{D_n\lambda_n}{f_n^{\text{loc}}}\le T_n^{\max}\\ & (\text{C8}):\ 0\le f_n^{\text{loc}}\le f_n^{\max}.\end{aligned}\tag{10}$$

Problem $(\mathcal{P}_3)$ can be further transformed into

$$\begin{aligned}(\mathcal{P}_4):\ \min_{f_n^{\text{loc}}}\ & \alpha D_n\lambda_n\left(f_n^{\text{loc}}\right)^2\\ \text{s.t.}\quad & (\text{C9}):\ \frac{D_n\lambda_n}{T_n^{\max}}\le f_n^{\text{loc}}\le f_n^{\max}.\end{aligned}\tag{11}$$

The closed-form solution to $(\mathcal{P}_4)$ can be obtained easily as follows.

1) If $([D_n\lambda_n]/T_n^{\max}) < f_n^{\max}$, local processing is feasible, and we have $x_n = 0$ and $f_n^{\text{loc}} = ([D_n\lambda_n]/T_n^{\max})$.

2) If $([D_n\lambda_n]/T_n^{\max}) = f_n^{\max}$, local processing is feasible, and we have $x_n = 0$ and $f_n^{\text{loc}} = ([D_n\lambda_n]/T_n^{\max}) = f_n^{\max}$.
3) If $([D_n\lambda_n]/T_n^{\max}) > f_n^{\max}$, local processing is infeasible, and we have $x_n = 1$.

Remark 3: Our local-related optimization is low in computational complexity, since the solution to offloading decision x_n and local computation resource allocation f_n^{loc} can be obtained easily in closed form.

Denote the set and number of local-processing IoT devices as $\mathcal{N}_{\text{loc}}$ and N_{loc}, respectively, which have been determined after local-related optimization is completed. For the IoT devices whose tasks are local infeasible, they may have the chance to offload their tasks for remote processing, and we denote the set and the number of local-infeasible IoT devices as $\mathcal{N}_1$ and N_1, respectively.

Remark 4: Except for device-side optimization, MEC server-side optimization will also contribute to capacity maximization by introducing NOMA, and contribute to energy saving maximization by performing admission control, user clustering and subcarrier allocation, and transmit power control optimization, which will be detailed as follows. Since this section is long, we consider it as a independent part for ease of understanding and clarity of structure.

V. Centralized MEC Server-Side Optimization

After offloading decision $\mathbf{x}$ and local computation resource allocation $\mathbf{f}^{\text{loc}}$ is obtained, problem $(\mathcal{P}_\mathbf{1})$ reduces to

$$\begin{aligned}(\mathcal{P}_\mathbf{5}):\ &\max_{\mathbf{y},\mathbf{p},\mathbf{S}}\ \sum_{n\in\mathcal{N}_1} E_n^{\text{save}}\\ \text{s.t.}\ &(\text{C1}):\ T_n \le T_n^{\max}\quad \forall n\in\mathcal{N}_1\\ &(\text{C2}):\ 0\le p_n \le p_n^{\max}\quad \forall n\in\mathcal{N}_1\\ &(\text{C3}):\ \sum_{k\in\mathcal{K}} s_{n,k} = 1\quad \forall n\in\mathcal{N}_1\\ &(\text{C4}):\ \sum_{n\in\mathcal{N}} s_{n,k} \le 2\quad \forall k\in\mathcal{K}\\ &(\text{C5}):\ s_{n,k}\in\{0,1\}\quad \forall n\in\mathcal{N}_1\quad \forall k\in\mathcal{K}\\ &(\text{C7}):\ y_n\in\{0,1\}\quad \forall n\in\mathcal{N}_1\end{aligned}\tag{12}$$

which is still hard to solve because of its combinatorial characteristics. Next, we will first decouple the access control strategies $\mathbf{y}$ from radio resource allocation and propose a low-complexity algorithm to solve it.

A. Admission Control Optimization

As was discussed, there can be at most $N_1 = 2K$ IoT devices be admitted in and served by the MEC server, so we need to perform admission control, to pick out the optimal $2K$ IoT devices, whose set is denoted as $\mathcal{N}_{\text{off}}$. The rest local-infeasible and not-admitted-in IoT devices will fail in task processing, whose set is denoted as $\mathcal{N}_{\text{fail}}$, and we have $\mathcal{N}_1 = \mathcal{N}_{\text{off}} \bigcup \mathcal{N}_{\text{fail}}$. Intuitively, IoT devices in good channel conditions will benefit from task offloading, since the data transmission rate will be high, and consequently less energy will be consumed in data transmission. Moreover, the characteristics of task themselves also play important roles. The tasks with more input data is not suitable for task offloading, since the energy consumed in data transmission may be higher than the energy saved in task processing; while tasks with high processing density can be beneficial in task offloading, since more energy will be saved in task processing. Inspired by the above observations, we define a criterion for admission control as follows:

$$G_n = g_n\lambda_n/D_n \tag{13}$$

where $g_n = \sum_{k\in\mathcal{K}} g_{n,k}$ indicates the general channel state of IoT device n. The larger g_n is, the more likely it is for IoT device n to perform task offloading. Based on the analysis, we propose an easy-to-implement algorithm, i.e., Algorithm 1, to capture admission control among the local-infeasible IoT devices in $\mathcal{N}_1$. We sort G_n in a decreasing order and select the first $2K$ IoT devices in $\mathcal{N}_1$ to perform task offloading, and the rest will fail in task processing, i.e., both local processing and computation offloading are infeasible for them.

Algorithm 1 Admission Control Algorithm

```
for $n \in \mathcal{N}_1$ do
    Calculate the value of $G_n$.
end for
Sort all the IoT devices in $\mathcal{N}_1$ in decreasing order and
    store these indices in the vector $I$.
if $N_1 > 2K$ then
    Set $y_n = 1$ for $n = I[k]$ where $k = 1, ..., 2K$, and $y_n = 0$
        with $k = 2K + 1, ..., N_1$.
else
    Set $y_n = 1$ for $n \in \mathcal{N}_1$.
end if
Set $\mathcal{N}_{off} = \{n|y_n = 1, n \in \mathcal{N}_1\}$.
Set $\mathcal{N}_{fail} = \{n|y_n = 0, n \in \mathcal{N}_1\}$.
```

B. User Clustering and Resource Allocation

After admission control $\mathbf{y}$ is obtained, problem $(\mathcal{P}_\mathbf{5})$ reduces to the joint optimization of user clustering and subcarrier allocation $\mathbf{S}$, and transmit power control $\mathbf{p}$, as follows:

$$\begin{aligned}(\mathcal{P}_\mathbf{6}):\ &\max_{\mathbf{p},\mathbf{S}}\ \sum_{n\in\mathcal{N}_{\text{off}}} \left[\alpha D_n\lambda_n\left(f_n^{\max}\right)^2 - E_n^{\text{mec}}\right]\\ \text{s.t.}\ &(\text{C1}):\ T_n^{\text{mec}} \le T_n^{\max}\quad \forall n\in\mathcal{N}_{\text{off}}\\ &(\text{C2}):\ 0\le p_n \le p_n^{\max}\quad \forall n\in\mathcal{N}_{\text{off}}\\ &(\text{C3}):\ \sum_{k\in\mathcal{K}} s_{n,k} = 1\quad \forall n\in\mathcal{N}_{\text{off}}\\ &(\text{C4}):\ \sum_{n\in\mathcal{N}} s_{n,k} \le 2\quad \forall k\in\mathcal{K}\\ &(\text{C5}):\ s_{n,k}\in\{0,1\}\quad \forall n\in\mathcal{N}_{\text{off}}\quad \forall k\in\mathcal{K}.\end{aligned}\tag{14}$$

In what follows, we will first reformulate problem $(\mathcal{P}_\mathbf{6})$ into an MWIS problem in the graph theory, and then we propose a low-complexity algorithm to solve it.

1) Feasibility Analysis of Given NOMA Cluster: Let $\mathcal{Q}$ represent an IoT device cluster, where all IoT devices share a same subcarrier. On account of the restraint (C4) in $(\mathcal{P}_\mathbf{6})$, the size of device cluster $\mathcal{Q}$ should be no more than 2. We then assign subcarrier k to $\mathcal{Q}$ and analyze the feasibility, i.e., whether the IoT devices in $\mathcal{Q}$ can meet their maximum tolerable latency requirements or not. Next, let $(\mathcal{Q}, k)$ denote a NOMA cluster, and the feasibility problem of $(\mathcal{Q}, k)$ can be given by

$$\text{find } \left\{\mathbf{p} | T_n^{\text{mec}} \le T_n^{\max}, 0 \le p_n \le p_n^{\max}, n \in \mathcal{N}_{\text{off}}\right\}. \qquad (15)$$

Based on (15), we could obtain the minimum transmit power $\bar{p}_n, n \in \mathcal{Q}$ of each IoT device n in cluster $\mathcal{Q}$. However, the obtained $\bar{p}_n$ may be infeasible, so we should verify whether $\bar{p}_n$ is less than or equal to the power budget $p_n^{\max}$. According to constraints (C2) and (C1) in $(\mathcal{P}_6)$, it can be known that in NOMA cluster $(\mathcal{Q}, k)$, the minimum transmit power $\bar{p}_n$ is achieved when $T_n^{\text{mec}} = T_n^{\max}, n \in \mathcal{Q}$. Substituting the expression of T_n^{mec} in (3) into $\bar{p}_n$, we have

$$\frac{D_n}{B_0 \log_2\left(1 + \frac{\bar{p}_n g_{n,k}}{\sum_{\substack{i \in \mathcal{Q} \\ g_{i,k} < g_{n,k}}} s_{i,k} g_{i,k} p_i + \sigma^2}\right)} + \frac{D_n \lambda_n}{f_n^{\text{mec}}} = T_n^{\max}. \qquad (16)$$

Rearranging it, we have

$$\bar{p}_n = \left(2^{\frac{D_n}{\left(T_n^{\max} - \frac{D_n \lambda_n}{f_n^{\text{mec}}}\right) B_0}} - 1\right)\left(\sum_{\substack{i \in \mathcal{Q} \\ g_{i,k} < g_{n,k}}} \frac{s_{i,k} \bar{p}_i g_{i,k}}{g_{n,k}} + \frac{\sigma^2}{g_{n,k}}\right). \qquad (17)$$

Perform the above process for each IoT device n in cluster $\mathcal{Q}$, we can obtain its minimum transmit power $\bar{p}_n, n \in \mathcal{Q}$. Then, by comparing $\bar{p}_n$ with $p_n^{\max}$ for each IoT device $n \in \mathcal{Q}$, we can know whether the current NOMA cluster $(\mathcal{Q}, k)$ is feasible for problem $(\mathcal{P}_6)$.

2) MWIS Reformulation: Problem $(\mathcal{P}_6)$ is similar with MWIS problems in several aspects.

1) In MWIS, two vertices must be nonadjacent in the graph, and similarly, in problem $(\mathcal{P}_6)$, two NOMA clusters cannot be allocated with the same subcarrier or contain the same IoT devices.
2) Moreover, the objective of problem $(\mathcal{P}_6)$ is to maximize the total energy saving of all devices, and similarly, the goal of MWIS is to maximize the weight of all vertices.

Therefore, the NOMA clusters can be considered to be the vertices in MWIS. Motivated by the observation, we concentrate on the MWIS-based methods. In what follows, we will introduce the detailed process of transforming problem $(\mathcal{P}_6)$ into the MWIS-based problem.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ represent a weighted undirected graph, where $\mathcal{V}$ stands for the set of all the vertices, $\mathcal{E}$ is the set of all the edges, and $\mathcal{W}$ denotes the set of vertex weights. Denote $v = (n_1^v, n_2^v, k^v)$ as a vertex in $\mathcal{V}$, which is the combination of two IoT devices n_1^v and n_2^v, and one subcarrier k^v, and satisfies $n_1^v \in \mathcal{N}_{\text{off}}, n_2^v \in \mathcal{N}_{\text{off}}, n_1^v \neq n_2^v, k^v \in \mathcal{K}$. Between two different vertices v_i and v_j, only when they contain the same IoT devices (any one or the both) or the same subcarrier, edge (v_i, v_j) exists. For notation simplicity, we define the *utility* of IoT device n as $U_n = \alpha D_n \lambda_n (f_n^{\max})^2 - E_n^{\text{mec}}$, based on which the weight of vertex v can be given by

$$W_v = U_{n_1^v}\left(p_{n_1^v}^*, p_{n_2^v}^*, k^v\right) + U_{n_2^v}\left(p_{n_1^v}^*, p_{n_2^v}^*, k^v\right) \qquad (18)$$

where $U_{n_1^v}$ and $U_{n_2^v}$ are the utility of two IoT devices n_1^v and n_2^v, respectively, which are determined by the optimal transmit power $p_{n_1^v}^*, p_{n_2^v}^*$, and the subcarrier k^v allocated to them.

Moreover, let $\mathcal{N}_\mathcal{G}(v)$ represent the neighborhood of vertex $v \in \mathcal{G}$, and define $\mathcal{N}_\mathcal{G}^+(v) = \mathcal{N}_\mathcal{G} \bigcup \{v\}$, we have the following definitions.

Definition 1 (Degree): The degree of vertex v means the number of its neighbors, and is represented by $d_\mathcal{G}(v)$.

Definition 2 [Independent Set (IS)]: $\mathcal{U}$ is termed as an IS of graph $\mathcal{G}$ if it satisfies: 1) $\mathcal{U} \subseteq \mathcal{G}$ and 2) $\forall v_i, v_j \in \mathcal{U}$, while $(v_i, v_j) \notin \mathcal{E}$.

Definition 3 [Maximum Weighted IS (MWIS)]: $\mathcal{U}$ is referred to as an MWIS of $\mathcal{G}$ if it satisfies: 1) $\mathcal{U}$ is an IS of graph $\mathcal{G}$ and 2) the total weight of all the vertices in $\mathcal{U}$ (i.e., $\sum_{v_i \in \mathcal{U}} W_{v_i}$) reaches the maximum among all ISs of $\mathcal{G}$.

Based on the definition of MWIS, we have the following theorem.

Theorem 1: Problem $(\mathcal{P}_6)$ can be equivalently transformed to the problem of determining the MWIS of $\mathcal{G}$.

Proof: For each vertex v, W_v is the maximum total utility, i.e., the utility under the optimal transmit power, of the two IoT devices inluded in v. Therefore, $\sum_{v_i \in \mathcal{U}} W_{v_i}$ is precisely the objective function of problem $(\mathcal{P}_6)$. Moreover, since a vertex is in fact a feasible NOMA cluster, constraints (C1) and (C2) hold. In addition, the independence between the vertices in MWIS $\mathcal{U}$ assures constraints (C3)–(C5) hold. Therefore, solving problem $(\mathcal{P}_6)$ is equivalent to finding out the MWIS in graph $\mathcal{G}$. ■

The MWIS problem is a classic NP-complete problem and hard to solve. In the following, we transform it into a MWIS problem in the graph theory [42]. Based on the characteristics of MWIS, we will propose an easy-to-implement heuristic algorithm (Algorithm 2) to solve $(\mathcal{P}_6)$. For easy understanding, we give some explanations about Algorithm 2 as follows.

Lines 1–19 is the graph initialization process, where we construct the weighted undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ as follows.

1) We start from subcarrier $k = 1$, and assume IoT device $n = 1$ is associated with it. Then, we calculate the minimum transmit power $\bar{p}_n$ according to (18), and verify the feasibility of $\bar{p}_n$ by comparing it with $p_n^{\max}$.
2) If device $n = 1$ is infeasible for subcarrier k, we suppose device $n = n + 1$ is associated with subcarrier k, and then continue to calculate $\bar{p}_n$ and judge the feasibility; otherwise, if device $n = 1$ is feasible on subcarrier $k = 1$, then we update the graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ by allocating IoT device $n = 1$ to subcarrier $k = 1$, and then we find the second IoT device for the $(n = 1, k = 1)$ pair.
3) Then, start with $j = n + 1$ (currently, $j = 2$), we traverse all the subsequent IoT devices. For $j = 2$, we first assume it is associated with the $(n = 1, k = 1)$ pair, and then we calculate the minimum transmit power of IoT devices $n = 1$ and $j = 2$, i.e., $\bar{p}_n$ and $\bar{p}_j$, and then evaluate the feasibility by comparing them with $p_n^{\max}$ and $p_j^{\max}$, respectively.
4) If infeasible, we let $j = j + 1 = 3$ and obtain $\bar{p}_n$ and $\bar{p}_j$, and then , we judge their feasibility; otherwise, we will

Algorithm 2 MWIS-Based User Clustering and Resource Allocation Algorithm

```
Initialization:
 1: $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})=\emptyset$.
 2: for $k=1:K$ do
 3:     Set $n=1$.
 4:     while $n\le 2K$ do
 5:         Calculate $\bar{p}_n$ according to (17).
 6:         if $\bar{p}_n\le p_n^{max}$ then
 7:             Update the graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$.
 8:             Set $j=n+1$.
 9:             while $j\le 2K$ do
10:                 Calculate $\bar{p}_n$ and $\bar{p}_j$ according to (17).
11:                 if $\bar{p}_n\le p_n^{max}$ & $\bar{p}_j\le p_j^{max}$ then
12:                     Update the graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$.
13:                 end if
14:                 $j=j+1$.
15:             end while
16:         end if
17:         $n=n+1$.
18:     end while
19: end for
20: Obtain the neighborhood $\mathcal{N}_{\mathcal{G}}(v)$ and the degree $d_{\mathcal{G}}(v)$ of
    each vertex $v\in\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$ based on the constructed
    graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$.
21: Let $\mathcal{U}=\emptyset$, $i=0$ and $\mathcal{G}_i(\mathcal{V},\mathcal{E},\mathcal{W})=\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$.
Iteration:
22: while $\mathcal{V}(\mathcal{G}_i)\neq\emptyset$ do
23:     Fine the set of vertexes $\mathcal{V}_1$ in $\mathcal{G}_i$ by the rule
        $$\mathcal{V}_1=\left\{v\middle|W_v\ge\sum_{u\in\mathcal{N}_{\mathcal{G}_i}^{+}(v)}\frac{W_u}{d_{\mathcal{G}_i(u)}+1}\right\}. \quad (19)$$
24:     Choose the vertex $v^*=\arg\max_{v\in\mathcal{V}_1}\frac{W_v}{d_{\mathcal{G}_i(v)}+1}$.
25:     Set $\mathcal{U}=\mathcal{U}\bigcup\{v^*\}$.
26:     Let $\mathcal{G}_{i+1}=\mathcal{G}_i\Big(\mathcal{V}(\mathcal{G}_i)-\mathcal{N}_{\mathcal{G}_i}^{+}(v^*)\Big)$.
27:     $i=i+1$.
28: end while
29: Output: The MWIS and get the corresponding $\mathbf{S}$ and $\mathbf{p}$.
```

update the graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$ by considering ($n=1$, $j=2$,$k=1$) as a NOMA cluster, i.e., a vertex in the graph.

5) Repeat the above process 1)–4) for all $j\in\mathcal{N}_{\text{off}}$, $j>n$, we can obtain all the feasible NOMA clusters ($n=1,j,k=1$),$j\in\mathcal{N}_{\text{off}}$, $j>n$.
6) Repeat the above process 1)–5) for all the $n\in\mathcal{N}_{\text{off}},j\in\mathcal{N}_{\text{off}}$ and for each k, $k\in\mathcal{K}$, we can obtain all the feasible NOMA clusters, i.e., the initial graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$.

Line 21 is the initial process before the subsequent iteration, where i is the iteration index, and $\mathcal{G}_i(\mathcal{V},\mathcal{E},\mathcal{W})$ is the graph in the ith iteration. In addition, set $\mathcal{U}$ will be used to store the elements of MWIS, i.e., the optimal joint user clustering, subcarrier allocation, and transmit power control scheme.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Number of IoT devices, N	30
Number of subcarriers, K	5
Cell radius	1000 m
Pathloss	d^{-3}
Shadowing	Log normal as $\mathcal{N}(0,8^2)$
Fading	Rayleigh fading with 1 variance
Subcarrier bandwidth, B_0	180 KHz
Noise power spectrum density	-174 dBm/Hz
Max transmit power, p_n^{max}	0.1 W
Input data size, D_n	0.4 ∼ 0.6 Kbit
Processing density, λ_n	100
MEC server capability, f^{mec}	3 G cycles/s
Idle power, p_n^{id}	$1mW$
Local capability constraint, f_n^{max}	0.05 G cycles/s
Maximum tolerable latency, T_n^{max}	10 ms
CPU architecture based parameter, α	10^{-27}

Lines 22–28 is the main execution process for finding the MWIS of graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{W})$. For notational simplicity, in this part, $\mathcal{G}_i(\mathcal{V},\mathcal{E},\mathcal{W})$ and $\mathcal{G}_{i+1}(\mathcal{V},\mathcal{E},\mathcal{W})$ are abbreviated to $\mathcal{G}_i$ and $\mathcal{G}_{i+1}$, respectively. In the ith iteration, when the set of vertex is nonempty, i.e., $\mathcal{V}(\mathcal{G}_i)\neq\emptyset$, we perform the following steps.

1) Among all the vertices in graph $\mathcal{G}_i$, we find the vertex v satisfying $W_v\ge\sum_{u\in\mathcal{N}_{\mathcal{G}_i}^{+}(v)}(W_u/[d_{\mathcal{G}_i(u)}+1])$, $v\in\mathcal{V}(\mathcal{G}_i)$, i.e., the vertex whose weight is greater than or equal to the average weight of its neighborhood, and put them in set $\mathcal{V}_1$.
2) Within set $\mathcal{V}_1$, we choose the vertex with the maximum average weight, which is denoted as v^*, i.e., $v^*=\arg\max_{v\in\mathcal{V}_1}(W_v/[d_{\mathcal{G}_i(v)}+1])$, and put v^* in the MWIS set $\mathcal{U}$.
3) Remove vertex v^* and all its neighbors from $\mathcal{G}_i$, and consider the rest vertices as the graph in the next iteration, i.e., $\mathcal{G}_{i+1}=\mathcal{G}_i(\mathcal{V}(\mathcal{G}_i)-\mathcal{N}_{\mathcal{G}_i}^{+}(v^*))$. The above steps 1)–3) keep repeating until the graph is empty, and the MWIS is obtained.

VI. SIMULATION RESULTS AND DISCUSSION

In this section, we provide simulations to verify the performance of our proposed joint optimization algorithm. We simulate a NOMA-MEC-enabled IoT network where an MEC server is deployed in the center, and multiple users are distributed uniformly within the cell. Detailed parameters are summarized in Table II.

We evaluate the performance of our proposed joint optimization algorithm, which is denoted as "Proposed" in the following, and we compare it with the following algorithms.

1) "Local," where all IoT devices process their tasks locally, and local resource allocation optimization is performed. When local processing is infeasible, unsuccessful task processing happens.
2) "All-offload," where all IoT devices offload their tasks to the MEC server, and server-side-related optimization is performed, including admission control, user clustering and subcarrier allocation, and transmit power control optimization. When server-side processing is infeasible, unsuccessful task processing happens.

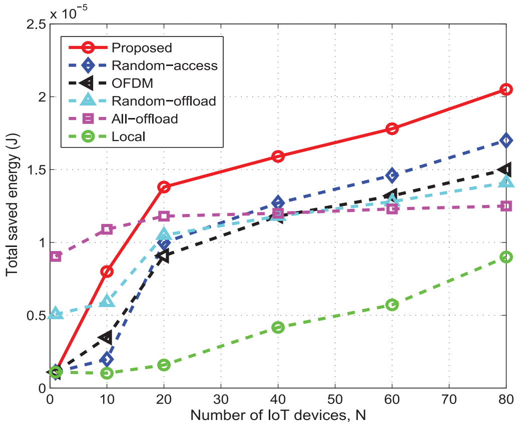

Fig. 1. Total saved energy versus the number of IoT devices N.

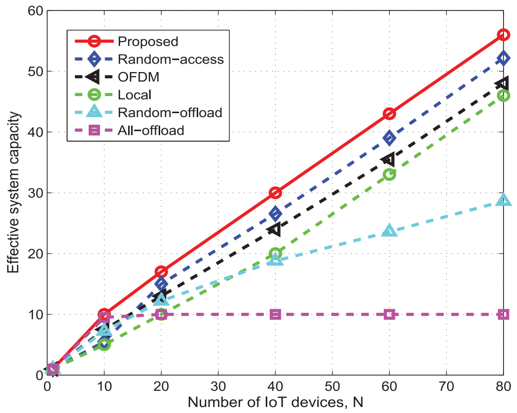

Fig. 2. Effective system capacity versus the number of IoT devices N.

3) "Random-offload," where the task-offloading decision is made randomly, and other optimization is performed.
4) "Random-access," where access control polices are made randomly, while other optimization is conducted.
5) "OFDM," where offloading decision making, local computation resource allocation, and access control optimization are executed the same as that in the proposed algorithm; the difference is that each subcarrier is allocated with a random IoT device, and transmit power control is performed according to (17).

Also, two performance metrics will be adopted as follows.

1) "Total saved energy," i.e., the objective in our formulated problem ($\mathcal{P}_1$), which means the saved energy of all IoT devices. As was mentioned, the saved energy of each IoT device is defined as the energy saving compared with local processing under the maximum local processing capability $f_n^{\max}$, i.e., $E_n^{\text{save}} = \alpha D_n \lambda_n (f_n^{\max})^2 - E_n$.
2) "Effective system capacity," i.e., the total number of IoT devices whose tasks are successfully processed.

In Fig. 1, we plot the total saved energy versus the total number of IoT devices N. When $N = 1$, for the algorithms with offloading decision optimization, including Proposed, Random-access and OFDM, the task will be processed by IoT device locally according to offloading decision optimization, so their saved energy is the same with Local; for All-offload, all the tasks will be offloaded, and for Random-offload, tasks may be offloaded randomly, so the energy savings of the two algorithms are much higher owing to task offloading. When N increases from 1 to 20, both local optimization and server-side optimization play their roles and thus to benefit some IoT devices, the total energy saving increases sharply thanks to the two aspects of optimization. When N continues to increase, the number of IoT devices is much more than that of subcarriers, so all the K subcarriers will be occupied by $N = 2K$ IoT devices and no extra device can benefit from task offloading. At this moment, the energy saving only comes from local computation resource allocation optimization, and therefore, increases relatively slow. Since All-offload can only benefit $N = 2K$ IoT devices, the total saved energy nearly keeps unchanged when N is greater than 20. As shown in Fig. 1, it can be seen that our proposed algorithm performs the best when N is sufficient enough, i.e., nearly more than 15, this is because both local-related and server-related optimizations come into full play.

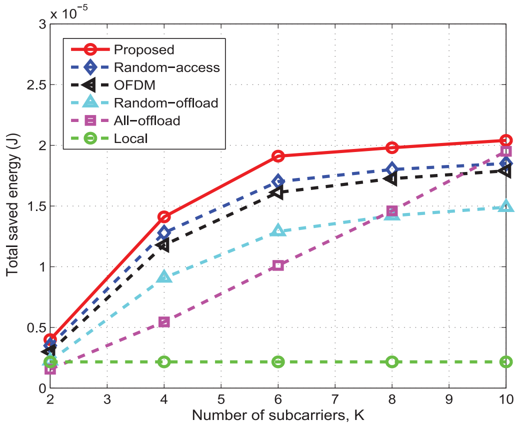

Fig. 3. Total saved energy versus the number of subcarriers K.

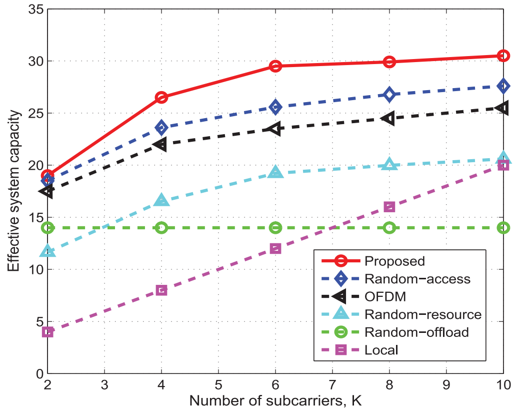

Fig. 4. Effective system capacity versus the number of subcarriers K.

In Fig. 2, we plot the effective system capacity versus the total number of IoT devices N. When N increases form 1 to 10, the subcarrier is relatively sufficient, so the system capacity grows relatively fast, and reachs 10 when $N = 10$ for Proposed and All-offload. When N is larger than 10, the effective system capacity of All-offload stops growing. When N continues to grow and is more than 20, the growth in effective system capacity mainly comes from local computation resource allocation optimization, and grows nearly linearly with the growth of N, where our Proposed algorithm always performs the best.

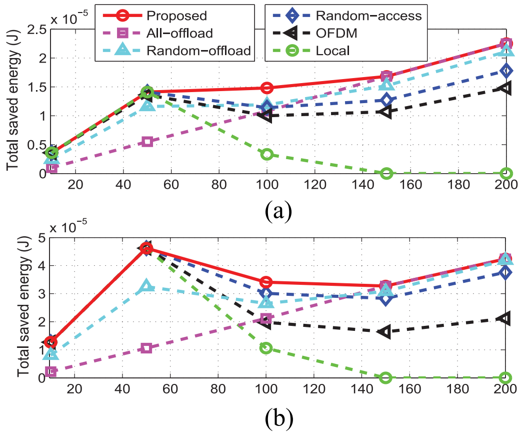

Fig. 5. Total saved energy versus processing density λ_n. (a) Processing density, λ_n, under $K = 5$ and $N = 30$. (b) Processing density, λ_n, under $K = 10$ and $N = 100$.

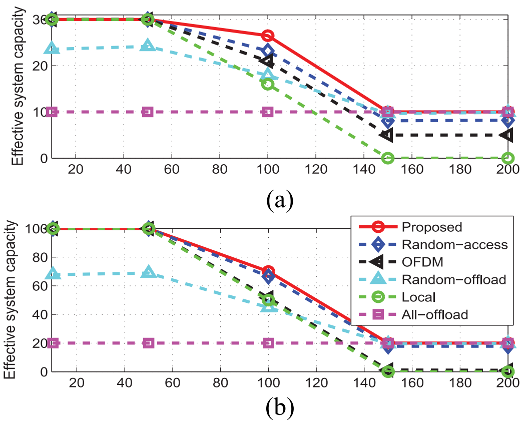

Fig. 6. blueEffective system capacity versus processing density λ_n. (a) Processing density, λ_n, under $K = 5$ and $N = 30$. (b) Processing density, λ_n, under $K = 10$ and $N = 100$.

In Figs. 3 and 4, we plot the total saved energy and the effective system capacity versus the total number of subcarriers K, respectively. As can be seen, the number of IoT devices that MEC server can afford increases linearly with K. In the All-offload method, because all IoT devices will offload their tasks if possible, both the energy consumption and the effective system capacity grow linearly with the growth of K. Since the number of channels K has no effect on local execution, the total saved energy consumption, and effective system capacity of Local remains unchanged. All other algorithms, including our Proposed, Random-offload, Random-access and OFDM, follow the same rules, i.e., at first their obtained energy saving and effective system capacity grow fast, and then gradually slower, with the number of K increases. Meanwhile, it can also be found that our Proposed algorithm always performs the optimum, followed by Random-access, OFDM, and Random-offload in sequence.

Figs. 5(a) and 6(a) plot the total saved energy and the effective system capacity versus the processing density λ_n, respectively, under default system size parameters, i.e., $K = 5, N = 30$. In order to demonstrate the scalability of our scheme, in Figs. 5(b) and 6(b), we plot the curves of our two metrics versus λ_n under $K = 10, N = 100$. It can be observed that the variance tendency of the two sets of curves in Fig. 5(a) and (b), and the two sets of curves in Fig. 6(a) and (b), follow almost exactly the same laws, which indicates our system can be scaled well. In the following, we take Figs. 5(a) and 6(a) as instances to analyze how the two metrics change under different processing density λ_n, respectively.

When λ_n falls among a tiny numerical interval [0, 50], i.e., the task is very simple and local processing is feasible and suitable, then all the 30 tasks will be processed locally with success. Except for Random-offload and All-offload, the effective system capacity of other algorithms all reach the maximum value 30 and have the similar energy-saving performance. For Random-offload, since some tasks are randomly offloaded, resulting in less saved energy and smaller effective system capacity, as shown in Figs. 5(a) and 6(a).

When λ_n grows to 100, the performance of local processing deteriorates rapidly, so the number of local feasible devices declines rapidly. At this point, task-offloading-related optimization starts working. However, observed from Fig. 5(a), $\lambda_n = 100$ is a critical point, which is neither sufficient big nor small, so the performance of local processing and task offloading are not very good. Consequently, for some algorithms, such as Random-access and OFDM, the performance in energy saving is even lower than that of $\lambda = 50$, as shown in Fig. 5(a). For the our Proposed, due to multiple-dimensional joint optimization, the energy saving does not decrease, but is only a bit improved. In terms of effective system capacity, since the number of local feasible devices drops greatly, and MEC server can only accommodate $2K = 10$ devices for task offloading, the effective capacity of all other algorithms reduce rapidly, except for All-offload, as shown in Fig. 6(a).

When $\lambda > 150$, local execution is not feasible and only the server-side optimization plays. Therefore, All-offload performs as well as our Proposed, while Local performs the worst with effective capacity and energy saving are both 0. Since the number of devices N is much more than the number of subcarriers K, the number of randomly offloaded IoT devices can always be more than $2K = 10$, so our Proposed and Random-offload algorithms can enable ten devices for successful task processing. Therefore, our Proposed and All-offload perform the best at this time, and the performance of the rest other algorithms locates between 10 and 0 as shown in Fig. 6(a).

Moreover, since the performance gain of All-offload only comes from offloading-related optimization, when the processing density λ increases, the task becomes more and more suitable for offloading, and thus, its energy saving grows linearly with λ increases, as shown in Fig. 5(a). However, no matter how λ varies, MEC server can always serve at most $2K = 10$ IoT devices under the current parameter settings, so the effective system capacity of All-offload is always 10 no matter how λ changes as shown in Fig. 6(a).

In Figs. 7 and 8, we plot the total saved energy and the effective system capacity versus the maximum tolerable latency $T_n^{\max}$, respectively. At first, when $T_n^{\max}$ is extremely small, i.e., the delay requirement is too strict, large quality of energy is needed for task processing. At this time, Local needs to work using all the local computation resource, however, it is still infeasible, making the energy saving and effective system

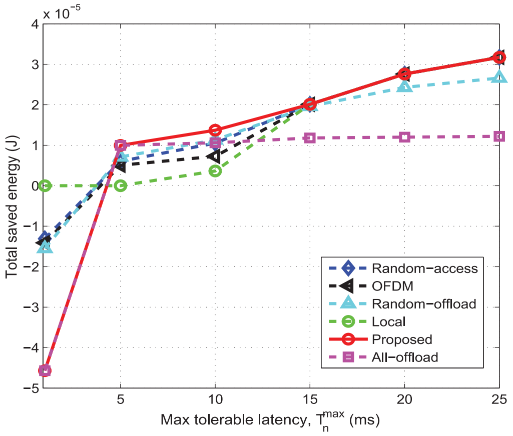

Fig. 7. Total saved energy versus max tolerable latency $T_n^{\max}$.

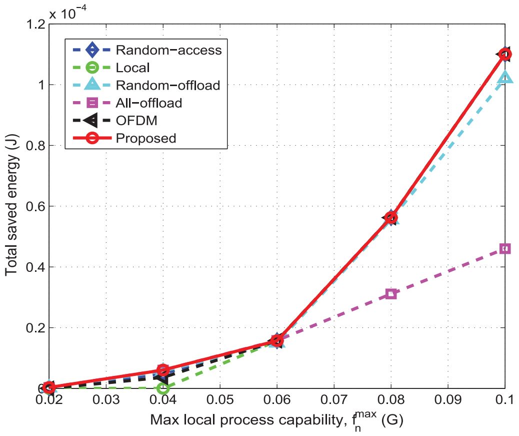

Fig. 9. Total saved energy versus max local process capability $f_n^{\max}$.

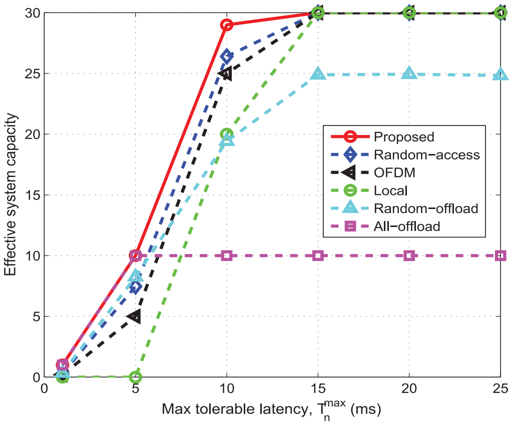

Fig. 8. Effective system capacity versus max tolerable latency $T_n^{\max}$.

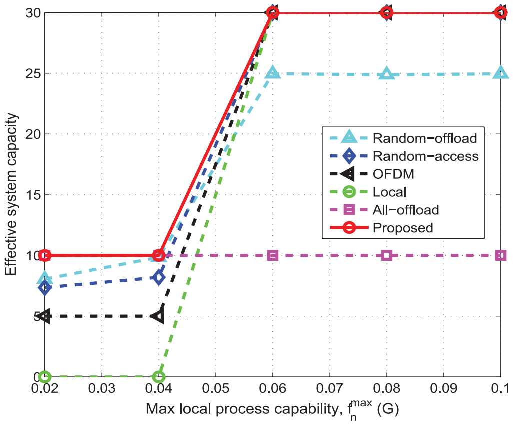

Fig. 10. Effective system capacity versus max local process capability $f_n^{\max}$.

capacity of Local are both 0. For other algorithms, according to offloading decision optimization, all IoT devices will offload their tasks, resulting in increased energy consumption. Nevertheless, task processing is still infeasible, so the energy saving is negative and the effective system capacity is also 0.

As $T_n^{\max}$ increases to 5 ms, task processing becomes feasible on the MEC server side, but local execution is still infeasible. Therefore, the energy saving and system capacity of Local are still 0, while other algorithms gradually have energy savings and system capacity growth.

As $T_n^{\max}$ grows to 15 ms, the number of local-feasible devices increases, so there's some performance gains for Local in both energy saving and system capacity increasing. Since All-offload can only benefit from task offloading, it can only benefit $N = 2K = 10$ IoT devices under default parameter settings. Our Proposed and other algorithms can benefit from both local-related and server-side optimizations, so the energy savings and system capacity increase rapidly.

When $T_n^{\max} >= 15$ ms, the delay requirement is very loose, and local execution can satisfy everyone's task processing requirement. As a result of offloading decision optimization, our Proposed, Random-offload, and OFDM will process all the tasks locally, so they could obtain the same quality energy saving and the maximum effective system capacity as Local does. What is more, it can also be known that the larger $T_n^{\max}$, the fewer needed local computation resource, and consequently the more energy savings, so the economized energy keeps growing as $T_n^{\max}$ grows for all the algorithms except for All-offload, since it can only serve $2K = 10$ IoT devices due to the constraints on subcarriers. It can also be observed form the above process that with task processing becomes feasible, our proposed algorithm always performs the best with respect to the two metrics.

In Figs. 9 and 10, we plot the total saved energy and the effective system capacity versus the maximum local processing capability $f_n^{\max}$, respectively. At the beginning, when $f_n^{\max}$ is small (between 0.02–0.04 G), local execution is infeasible, so the effective system capacity of Local is 0. Since Local will run on $f_n^{\max}$ as a result of local computation resource allocation, there is no energy saving in Local. At this time, the best method is task offloading, however, only $2K = 10$ IoT devices can be served successfully by the MEC server due to access control. It can be known from the two figures that only our Proposed and All-offload can reach the maximum effective system capacity, and save the most quality of energy. Other algorithms (including Random-offload, Random-access, and OFDM) that involve local-related optimization perform not as good as our Proposed and All-offload both in effective capacity and energy saving maximization.

When $f_n^{\max}$ increases from 0.04 to 0.06 G, local processing becomes feasible on some devices at this time. Therefore, in addition to the $2K = 10$ successful offloaded devices, local

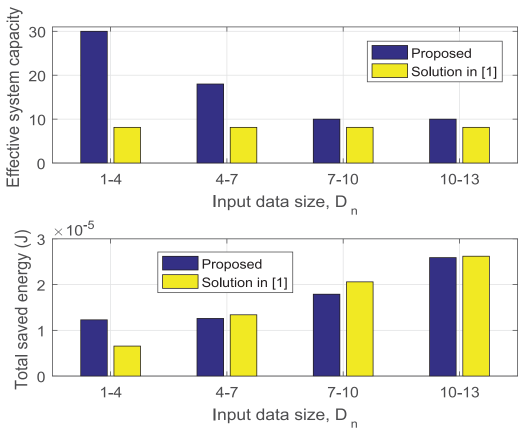

Fig. 11. Total saved energy and effective system capacity comparison under different input data size D_n.

execution also brings gains in energy efficiency improving and effective system capacity maximization, so the number of IoT devices increases rapidly and reaches the maximum value when $f_n^{\max} = 0.06$ G. However, in the process, the total energy saving grows not so rapid as the system capacity does, since the saved energy of each IoT device is defined as $E_n^{\text{save}} = \alpha D_n \lambda_n (f_n^{\max})^2 - E_n = \alpha D_n \lambda_n [(f_n^{\max})^2 - (f_n^{\text{loc}})^2]$ in local processing, and the gap between $f_n^{\max}$ and f_n^{loc} is not so large at this time.

When $f_n^{\max}$ $>=$ 0.06 G, the local processing capability is strong enough, and local execution is feasible for all the IoT devices. Therefore, except for Random-offload and All-offload, all IoT devices will process their tasks locally by offloading decision optimization in all other algorithms, and the effective system capacity reaches the maximum value 30 all the time. In Random-offload, some devices will offload their tasks randomly, but not all the offloaded tasks can be served by the wireless network due to admission control, the effective system capacity of Random-offloading is less than 30. At this region, with the increase of $f_n^{\max}$, the gap between $f_n^{\max}$ and f_n^{loc} becomes larger, and the energy saving grows faster and faster following quadratic functions.

In the above process of the growth of $f_n^{\max}$, our Proposed algorithm always performs the best in both effective system capacity maximization and energy saving optimization. Since in All-offload, all the IoT devices could not process their tasks locally, and among them only $2K$ devices will be served by task offloading, so the effective system capacity always equals to $2K = 10$. However, since $E_n^{\text{save}} = \alpha D_n \lambda_n (f_n^{\max})^2 - E_n^{\text{mec}}$, E_n^{save} of All-offload will also increase with $f_n^{\max}$ grows, although All-offload does not involve local-related optimization.

To further demonstrate the performance of our proposed algorithm, we compare it with the solution in [1] under different input data size D_n as shown in Fig. 11. In [1], there is no offloading decision making and local computation resource allocation optimization, so it could not be able to benefit from local-related optimization. What is more, there is no access control optimization in [1], and we perform random access control for comparison. In addition, the only objective of this method is to minimize the total system energy consumption, which is somewhat different with ours, since in our algorithm, the first objective is to maximize the system capacity, and then the total saved energy.

In the first subfigure in Fig. 11, we compare the effective system capacity between the two algorithms with the increase of D_n. When D_n is small, the processing amount $C_n = D_n \lambda_n$ is also small, so all tasks can be processed successfully by IoT devices themselves, and therefore, our algorithm could gain the maximum system capacity through offloading decision optimization, i.e., by letting all IoT devices perform local processing. While for the solution in [1], all IoT devices offload their tasks to MEC server, so it could only benefit from offloading-related optimization, including user clustering, subcarrier allocation, and power control. Whereas, constrained by wireless subchannels, the effective system capacity of [1] is much smaller than our proposed algorithm. With the increase of D_n, local processing becomes infeasible for some tasks, leading to a decrease in system capacity for our proposed algorithm, which is however, still higher than [1] owing to access control and local-related optimization.

In the bottom subfigure of Fig. 11, we plot the total saved energy consumption about the two algorithms. As was analyzed, when D_n is small, local processing is the best choice, and our algorithm could save more energy than [1] does by offloading decision optimization. When D_n grows, C_n also increases, then local processing becomes infeasible, and offloading-related optimizations gradually play key roles than local-related optimizations. From the subfigure, it can be known that our proposed algorithm performs a little worse than [1] in energy reduction, or, saved energy maximization, when D_n is not very small. This is because, we give priority to the effective system capacity, and then to the total saved energy maximization, as was analyzed in Remark 2, while the solution in [1] only intends to minimize the energy consumption, so it performs a little better than our proposed algorithm in energy saving maximization.

VII. Conclusion

In this article, we have considered the effective system capacity and energy saving maximization in a NOMA-MEC-based IoT system, by a joint optimization of the offloading decision making, local computation resource allocation, access control, user clustering, subcarrier assignment, and transmit power control. Employing distributed local related optimization, MWIS in graph theory, and heuristic algorithm, we have proposed low complexity algorithms to solve it. We have also presented abundant simulation results to demonstrate that our proposed joint optimization algorithm performs well in both effective system capacity optimization and energy saving maximization.

References

[1] A. Kiani and N. Ansari, "Edge computing aware NOMA for 5G networks," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1299–1306, Apr. 2018.

[2] C. Chen, J. Hu, T. Qiu, M. Atiquzzaman, and Z. Ren, "CVCG: Cooperative V2V-aided transmission scheme based on coalitional game for popular content distribution in vehicular ad-hoc networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 12, pp. 2811–2828, Dec. 2019.

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

[4] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[5] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[6] D. Zhai, R. Zhang, L. Cai, B. Li, and Y. Jiang, "Energy-efficient user scheduling and power allocation for NOMA-based wireless networks with massive iot devices," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1857–1868, Jun. 2018.

[7] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Netw. Appl.*, pp. 1–24, Jul. 2020.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[9] X. Yuan, H. Tian, H. Wang, H. Su, J. Liu, and A. Taherkordi, "Edge-enabled WBANs for efficient qos provisioning healthcare monitoring: A two-stage potential game-based computation offloading strategy," *IEEE Access*, vol. 8, pp. 92718–92730, 2020, doi: 10.1109/ACCESS.2020.2992639.

[10] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31–37, Dec. 2017.

[11] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang, and X. Chu, "MEC-assisted immersive VR video streaming over terahertz wireless networks: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9517–9529, Oct. 2020.

[12] J. Feng, F. R. Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6214–6228, Jul. 2020.

[13] Y. Wu, L. P. Qian, K. Ni, C. Zhang, and X. Shen, "Delay-minimization nonorthogonal multiple access enabled multi-user mobile edge computation offloading," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 392–407, Jun. 2019.

[14] X. Li, J. Li, Y. Liu, Z. Ding, and A. Nallanathan, "Residual transceiver hardware impairments on cooperative NOMA networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 680–695, Jan. 2020.

[15] X. Li, M. Zhao, Y. Liu, L. Li, Z. Ding, and A. Nallanathan, "Secrecy analysis of ambient backscatter NOMA systems under I/Q imbalance," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12286–12290, Oct. 2020, doi: 10.1109/TVT.2020.3006478.

[16] L. Dai, B. Wang, Y. Yuan, S. Han, I. Chih-Lin, 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.

[17] Z. Ding, D. W. K. Ng, R. Schober, and H. V. Poor, "Delay minimization for NOMA-MEC offloading," *IEEE Signal Process. Lett.*, vol. 25, no. 12, pp. 1875–1879, Dec. 2018.

[18] Y. Zhang, J. Ge, and E. Serpedin, "Performance analysis of a 5G energy-constrained downlink relaying network with non-orthogonal multiple access," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 8333–8346, Dec. 2017.

[19] Z. Ding, J. Xu, O. A. Dobre, and H. V. Poor, "Joint power and time allocation for NOMA-MEC offloading," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6207–6211, Jun. 2019.

[20] F. Fang, Y. Xu, Z. Ding, C. Shen, M. Peng, and G. K. Karagiannidis, "Optimal resource allocation for delay minimization in NOMA-MEC networks," *IEEE Trans. Commun.*, early access, Aug. 28, 2020, doi: 10.1109/TCOMM.2020.3020068.

[21] F. Fang, Y. Xu, Q.-V. Pham, and Z. Ding, "Energy-efficient design of IRS-NOMA networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 14088–14092, Nov. 2020.

[22] Y. Pan, M. Chen, Z. Yang, N. Huang, and M. Shikh-Bahaei, "Energy-efficient NOMA-based mobile edge computing offloading," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 310–313, Feb. 2019.

[23] F. Wang, J. Xu, and Z. Ding, "Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2450–2463, Mar. 2019.

[24] Z. Song, Y. Liu, and X. Sun, "Joint radio and computational resource allocation for NOMA-based mobile edge computing in heterogeneous networks," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2559–2562, Dec. 2018.

[25] M. Zeng and V. Fodor, "Energy-efficient resource allocation for NOMA-assisted mobile edge computing," in *Proc. IEEE PIMRC*, Bologna, Italy, Sep. 2018, pp. 1794–1799.

[26] X. Li, Y. Liu, H. Ji, H. Zhang, and V. C. M. Leung, "Optimizing resources allocation for fog computing-based Internet of Things networks," *IEEE Access*, vol. 7, pp. 34907–64922, 2019.

[27] L. P. Qian, A. Feng, Y. Huang, Y. Wu, B. Ji, and Z. Shi, "Optimal SIC ordering and computation resource allocation in MEC-aware NOMA NB-IoT networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2806–2816, Apr. 2019.

[28] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, "NOMA-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12244–12258, Dec. 2018.

[29] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12137–12151, Dec. 2018.

[30] Z. Ding, P. Fan, and H. V. Poor, "Impact of non-orthogonal multiple access on the offloading of mobile edge computing," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 375–390, Jan. 2019.

[31] X. Diao, J. Zheng, Y. Wu, and Y. Cai, "Joint computing resource, power, and channel allocations for D2D-assisted and NOMA-based mobile edge computing," *IEEE Access*, vol. 7, pp. 9243–9257, 2019.

[32] Z. Wei and H. Jiang, "Optimal offloading in fog computing systems with non-orthogonal multiple access," *IEEE Access*, vol. 6, pp. 49767–49778, 2018.

[33] X. Wen, H. Zhang, H. Zhang, and F. Fang, "Interference pricing resource allocation and user-subchannel matching for NOMA hierarchy fog networks," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 467–479, Jun. 2019.

[34] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1079–1092, Feb. 2019.

[35] S. Bi, L. Huang, and Y.-J. A. Zhang, "Joint optimization of service caching placement and computation offloading in mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4947–4963, Jul. 2020.

[36] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.

[37] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[38] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4321–4334, Jun. 2020.

[39] H. Cao, S. Wu, G. S. Aujla, Q. Wang, L. Yang, and H. Zhu, "Dynamic embedding and quality of service-driven adjustment for cloud networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1406–1416, Feb. 2020.

[40] H. Cao, A. Xiao, Y. Hu, P. Zhang, S. Wu, and L. Yang, "On virtual resource allocation of heterogeneous networks in virtualization environment: A service oriented perspective," *IEEE Trans. Netw. Sci. Eng.*, early access, Feb. 10, 2020, doi: 10.1109/TNSE.2020.2972602.

[41] D. Zhai, H. Li, X. Tang, R. Zhang, Z. Ding, and F. R. Yu, "Height optimization and resource allocation for NOMA enhanced UAV-aided relay networks," *IEEE Trans. Commun.*, early access, Nov. 16, 2020, doi: 10.1109/TCOMM.2020.3037345.

[42] J. Bondy and U. Murty, *Graph Theory*. Heidelberg, Germany: Springer, 2008.

Jianbo Du (Member, IEEE) received the Ph.D. degree in communication and information systems from Xidian University, Xi'an, China, in 2018.

She is currently a Lecturer with the Department of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, China. Her research interests include mobile edge computing, resource management, NOMA, vehicular networks, convex optimization, heuristic algorithms, and artificial intelligence and their applications in wireless communications.

Wenhuan Liu received the B.S. degree in communication engineering from Xi'an University of Posts and Telecommunications, Xi'an, China, in 2020, where he is currently pursuing the M.S. degree in communication and information systems.

His research interests include mobile-edge computing, resource management, NOMA, and their applications in wireless communications.

Guangyue Lu received the Ph.D. degree from Xidian University, Xi'an, China, in 1999.

From September 2004 to August 2006, he was a Guest Researcher with the Signal and Systems Group, Uppsala University, Uppsala, Sweden. Since 2005, he has been a Professor with the Department of Telecommunications Engineering, Xi'an University of Posts and Telecommunications, Xi'an. His current research area is in signal processing in communication systems, cognitive radio, and spectrum sensing.

Dr. Lu received the Award from the Program for New Century Excellent Talents in University, Ministry of Education, China, for his excellent contributions in education and research in 2009.

Jing Jiang (Member, IEEE) received the M.Sc. degree from Xidian University, Xi'an, China, in 2005, and the Ph.D. degree in information and communication engineering from Northwestern Polytechnic University, Xi'an, in 2009.

She was a Researcher and a Project Manager with ZTE Corporation, Shenzhen, China, from 2006 to 2012. She is currently a Professor with the Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications, Xi'an. Her research interests include massive multiple-input–multiple-output systems and millimeter-wave communications.

Prof. Jiang has been a member of 3GPP.

Daosen Zhai (Member, IEEE) received the B.E. degree in telecommunication engineering from Shandong University, Weihai, China, in 2012, and the Ph.D. degree in communication and information systems from Xidian University, Xi'an, China, in 2017.

He is currently an Assistant Professor with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an. His research interests focus on radio resource management in B5G and 6G, massive access techniques, air-and-ground-integrated network, and convex optimization and graph theory and their applications in wireless communications.

F. Richard Yu (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 2003.

From 2002 to 2006, he was with Ericsson, Lund, Sweden, and a start-up in California, USA. In 2007, he joined Carleton University, Ottawa, ON, Canada, where he is currently a Professor. His research interests include wireless cyber–physical systems, connected/autonomous vehicles, security, distributed ledger technology, and deep learning.

Dr. Yu received the IEEE Outstanding Service Award in 2016, the IEEE Outstanding Leadership Award in 2013, the Carleton Research Achievement Award in 2012, the Ontario Early Researcher Award (formerly, Premiers Research Excellence Award) in 2011, the Excellent Contribution Award at IEEE/IFIP TrustCom 2010, the Leadership Opportunity Fund Award from Canada Foundation of Innovation in 2009, and the Best Paper Awards at IEEE ICNC 2018, VTC 2017 Spring, ICC 2014, Globecom 2012, IEEE/IFIP TrustCom 2009, and International Conference on Networking 2005. He serves on the editorial boards of several journals, including the Co-Editor-in-Chief for *Ad-Hoc and Sensor Wireless Networks*, the Lead Series Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, and IEEE COMMUNICATIONS SURVEYS & TUTORIALS. He has served as the technical program committee co-chair of numerous conferences. He is a registered Professional Engineer in the province of Ontario, Canada, and a Fellow of the Institution of Engineering and Technology. He is a Distinguished Lecturer, the Vice President (Membership), and an Elected Member of the Board of Governors of the IEEE Vehicular Technology Society.

Zhiguo Ding (Fellow, IEEE) received the B.Eng. degree in electrical engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2000, and the Ph.D. degree in electrical engineering from Imperial College London, London, U.K., in 2005.

From July 2005 to April 2018, he was working with Queen's University Belfast, Belfast, U.K.; Imperial College; Newcastle University, Newcastle upon Tyne, U.K.; and Lancaster University, Lancashire, U.K. Since April 2018, he has been with the University of Manchester, Manchester, U.K., as a Professor of Communications. From October 2012 to September 2018, he has also been an Academic Visitor with Princeton University, Princeton, NJ, USA. His research interests are 5G networks, game theory, cooperative and energy harvesting networks, and statistical signal processing.

Dr. Ding received the best paper award in IET ICWMC-2009 and IEEE WCSP-2014, the EU Marie Curie Fellowship from 2012 to 2014, the Top IEEE TVT Editor in 2017, the IEEE Heinrich Hertz Award in 2018, and the IEEE Jack Neubauer Memorial Award in 2018. He is serving as an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and *Journal of Wireless Communications and Mobile Computing*, and was an Editor for IEEE WIRELESS COMMUNICATION LETTERS and IEEE COMMUNICATION LETTERS from 2013 to 2016.