Received July 28, 2019, accepted August 10, 2019, date of publication August 20, 2019, date of current version September 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936435

Energy Efficiency Based Joint Computation Offloading and Resource Allocation in Multi-Access MEC Systems

XIAOTONG YANG[1,2], XUEYONG YU[1,2], HAO HUANG[3,4], (Student Member, IEEE), AND HONGBO ZHU[1,2]

[1]Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[2]Engineering Research Center of Health Service System Based on Ubiquitous Wireless Networks, Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[3]Key Lab of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[4]National Engineering Research Center for Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Corresponding authors: Hongbo Zhu (zhuhb@njupt.edu.cn) and Xueyong Yu (yuxy@njupt.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61871446, and in part by the Open Project of Wireless Communication Key Lab of Jiangsu Province under Grant ZS002NY17013-2017WICOM03.

ABSTRACT With the rapid growth of computation demands from mobile applications, mobile-edge computing (MEC) provides a new method to meet requirement of high data rate and high computation capability. By offloading the latency-critical or computation-intensive tasks to the edge server, mobile devices (MDs) could save energy consumption and extend battery life. However, unlike cloud servers, resource bottlenecks in MEC servers limit the scalability of offloading. Hence, computation offloading and resource allocation need to be optimized. Toward this end, we consider a multi-access MEC servers system in which Orthogonal Frequency-Division Multiplexing Access (OFDMA) is used as the transmission mechanism for uplink. In order to minimize energy consumption of MDs, we propose a joint optimization strategy for computation offloading, subcarrier allocation, and computing resource allocation, which is a mixed integer non-linear programming (MINLP) problem. First, we design a bound improving branch-and-bound (BnB) algorithm to find the global optimal solution. Then, we present a combinational algorithm to obtain the suboptimal solution for practical application. Simulation results reveal that the combinational algorithm performs very closely to the BnB algorithm in energy saving, but it has a better performance in average algorithm time. Furthermore, our proposed solutions outperform other benchmark schemes.

INDEX TERMS Mobile-edge computing, multi-access edge computing, computation offloading, resource allocation.

I. INTRODUCTION

With the popularity of smart mobile devices (MDs) like intelligent mobile phones, smart watch/band and Internet of Things (IoT) devices such as shared power supply, and shared bike, lots of new mobile applications come with the tide of fashion [1], [2]. These new mobile applications, e.g., e-Health care, face recognition, surveillance, and augmented reality/virtual reality (AR/VR), are not only computation-intensive but high energy consumption [3]. However, this unparalleled growth does not match the improvement on MDs' batteries and computation capacity. Given the tremendous increase in the usage of the MDs, mobile-edge computing (MEC) can bridge the gap between restricted capabilities of MDs and increasing computing demand [4]. MEC performs computation-intensive tasks instead of MDs by collecting a large amount of idle resources and storage space distributed at the edge of the network, so as to meet the strict delay requirements [5], [6].

Nevertheless, offloading generates additional overhead because of the communication between the MDs and the MEC servers, and as a result, offloading strategy is particularly important. In [7], the authors studied the offloading decision among multiple devices and one MEC server.

The associate editor coordinating the review of this article and approving it for publication was Yue Cao.

For Ultra-Dense Networks in future 5G network, [8] considered a multi-access MEC scenario and proposed a heuristic greedy offloading scheme to solve computation offloading problem. However, both wireless and computing resources affect the performance of offloading strategy. Because the former affects the data transmission rate and the MDs' energy consumption, and the latter influences the task computing delay [9].

Currently, a great deal of existing works on MEC just focus on the computation offloading decision or the joint wireless and computing resources optimization problem, and deep learning techniques are also being used to solve these problems [10], [11]. Recently, joint offloading decision and resource optimization have been taken into account [12], [13]. In the case of multi-MD scenario, the authors focused on the offloading strategy problem along with the power and computation resources optimization in [14]. Reference [15] was centered on the offloading decision, wireless resource optimization, and computational resource optimization to achieve energy saving. In multi-MEC servers and multi-MD scenario, MDs need not only to decide whether to offload but also to decide where to offload. Paper [16] considered an signal-access MEC server system with the transmission mechanism of Orthogonal Frequency-Division Multiplexing Access (OFDMA) and optimized offloading strategy and radio resource allocation. In [17], genetic algorithm was used for solving the joint optimization problem in the small cell network (SCN) to find a suboptimal result. Reference [18] optimized the offloading utility, that is, the weighted sum of time and energy consumption of each user in the scenario of multi-user and multi-server. In the paper [19], the author studied three kinds of computation offloading strategies in the distributed MEC system, namely local computing, offloading tasks to local regional servers and offloading to nearby regional servers, and solved the problem in the short term through the idea of game theory. In [20], the author studied bits resource allocation in multi-user MEC offloading system to minimize the weighted sum of energy consumption. Paper [21] solved the problem of joint computation offloading and user association in the multi-task MEC system, where the allocation of computation resources and transmission power is also considered, so as to save the overall energy consumption of the system. In [22], the author proposed a distributed optimization algorithm in heterogeneous networks with MEC to minimize mobile terminals' cost, where joint optimized the computation offloading, subchannel allocation, power allocation and CPU-cycle allocation.

In this paper, we consider a MEC system with multiple MEC servers serving multiple MDs, and propose optimal and sub-optimal algorithms for the joint optimization of the offloading decision, wireless resources allocation, and computation resources allocation to minimize the energy consumption of MDs under the constraint of delay, so that save energy for devices. The main contributions of this paper are listed as follows:

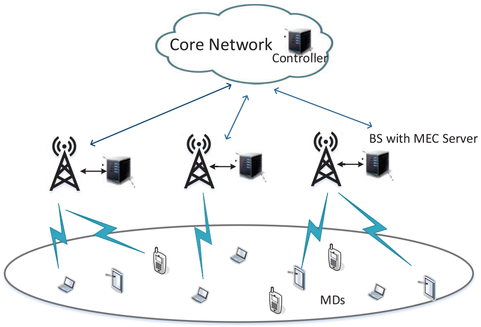

FIGURE 1. System architecture.

- We come up with an OFDMA based MEC system model with multi-access BS and limited resource. From the perspective of energy saving, we study the problem of joint optimization offloading strategy, wireless resource allocation, and computational resource allocation.
- In the above system, the original problem which has three mutually constrained variables is a mixed integer non-linear programming MINLP) problem, and still non-convex even if the continuous relaxation is adopted. Because of the mutual constraints between variables, it is difficult to decompose this problem into sub-problems. We transform the problem into mixed integer non-linear convex problem by variable combination, which is relatively easy to be solved. Then, the bound improving branch-and-bound (BnB) algorithm is used to get the global optimal solution.
- Consider the complexity and average algorithm time of the BnB algorithm, we propose an intelligent heuristic algorithm. Since the heuristic algorithm cannot always find the solution to the problem, we further combine the heuristic algorithm with BnB algorithm to get the low-complexity suboptimal combination algorithm for practical application.
- In the simulation process, we use different system configurations to testify the validity of our proposed algorithm. It is worth mentioning that the performance of the combinational algorithm is very closely to the optimal BnB algorithm, but the former has a better performance in terms of time saving. In addition, the performance of our proposed algorithm are better than that of baseline algorithms.

Article Organization: The rest of this paper is organized as follows. In Sect. II, we propose the system model. In Sect. III, the joint computation offloading and resource allocation problem is formulated, and the problem transformation is presented. In Sect. IV, we introduce our proposed algorithms. Simulation results are showed in Sect. V. Finally, we conclude the article in Sect. VI.

II. SYSTEM MODEL

Consider a scenario of multi-access MEC system with multiple MDs and multiple MEC servers shown in Fig. 1.

Given the small coverage of the base station (BS), we use OFDMA as the uplink transmission mechanism, in which interference among MDs could be ignored, with subcarriers as the radio resource. In this snapshot, each BS deploys an available MEC server with limited computation resource, and each BS is connected to the core network. Generally speaking, MEC servers are physical server or virtual machine with certain computation capacity. Each MD can offload the task to a nearby BS that can be connected to for computing or performs computation locally. The S MEC servers, which work independently, sever K MDs through N available subcarriers for uplink wireless transmission. Let $\mathcal{S} = \{1, 2, \ldots, S\}$, $\mathcal{N} = \{1, 2, \ldots, N\}$ and $\mathcal{K} = \{1, 2, \ldots, K\}$. We assume that the central processing units (CPUs) of the MEC servers are unoccupied at the present time.

A. MDs LOCAL COMPUTATION

We start coping with the case in which the tasks be completed locally. Each MD k has a computation task and it can be described by a three-field notation $\Psi_k = (D_k, X_k, \tau_k)$. D_k is the task input-data size (in bits), including system settings, program codes, and so on. X_k denotes the computation workload/intensity (in CPU cycles per bit), which may differ from different applications. τ_k is maximum accomplished deadline(in seconds). So we can figure out that $D_k X_k$ is the required CPU cycles for completing the task k. Assume that the tasks cannot be divided into subtasks. Besides, these above parameters that are related to the nature of the applications can be estimated by applying program profilers (e.g., as in [16]).

F_k^l denotes the frequency of local CPU, i.e., the local computation capability of MD k. T_k^l is the time cost for local computation, thus the local computation time for MD k can be given as

$$T_k^l = \frac{D_k X_k}{F_k^l}. \tag{1}$$

According to [23], the energy consumption on MD k during each CPU cycle is a super-linear function of execution frequency and is written as

$$p_k^l = k_0 (F_k^l)^2, \tag{2}$$

where k_0 is a constant related to the CPU of MDs, and typically, $k_0 = 1 \times 10^{-24}$ [23].

Accordingly, the local energy computation of MD k is expressed as

$$E_k^l = k_0 (F_k^l)^2 D_k X_k. \tag{3}$$

Since the T_k^l and E_k^l are determined only by F_k^l, D_k and X_k, the computation task are known.

B. TASK UPLOADING

By offloading the task to an MEC server, the MD would save energy on task computation. At the same time, it would consume extra time and energy for uploading the task.

A representative remote computation process consists of the following three parts.

1) The MD k uploads one computation task $A(D_k, X_k, \tau_k)$ to the MEC server s through uplink subcarriers.

2) The MEC server s executes the task k and allocates $F_{k,s}$ computation resource to it.

3) The MEC server s transmits output data back to the MD.

Here, the overhead of the output data is ignored in the last part as in [14], since the amount of it is generally much smaller than that of the input data.

For the OFDMA mechanism, interference is ignored on account of the exclusive subcarrier allocation. Therefore the data rate can be written as

$$R_{k,s}(W) = B_N \sum_{n \in \mathcal{N}} w_{k,n,s} \log_2 (1 + \frac{g_{k,n,s} P_k}{\sigma^2}), \tag{4}$$

where B_N (Hz) is the bandwidth of each subcarrier and σ^2 is the background noise variance. $W = \{w_{k,n,s} \mid w_{k,n,s} \in \{0, 1\}, k \in \mathcal{K}, n \in \mathcal{N}, s \in \mathcal{S}\}$ denotes the subcarrier allocation matrix, identifying whether the subcarrier n is assigned to the MD k and corresponding MEC server s. $g_{k,n,s}$ is the channel gain between the MEC server s and MD k on the subcarrier n and we let $G = \{g_{k,n,s}, k \in \mathcal{K}, n \in \mathcal{N}, s \in \mathcal{S}\}$ denote the channel gain matrix. P_k denotes the transmission power, and it can be allocated by the MDs. We define the maximum transmission power as P^m. Apparently, any power optimization solution have a good impact on system performance. For the sake of simplicity, P_k remains at a random level in this paper.

C. MEC REMOTE COMPUTATION

The total remote computation completion time cost for MD k, i.e.,$T_k^r(W, F)$, is given by

$$T_k^r(W, F) = T_k^t(W) + T_k^e(F), \tag{5}$$

where the $T_k^t(W)$ and $T_k^e(F)$ are the uplink transmission time and remote execution time for MD k, repectively [24]. $T_k^t(W)$ can be written as

$$T_k^t(W) = \frac{D_k}{R_{k,s}(W)}. \tag{6}$$

The remote execution time $T_k^e(F)$ can be obtained as

$$T_k^e(F) = \frac{D_k X_k}{f_{k,s}}, \tag{7}$$

where $F = \{f_{k,s}, k \in \mathcal{K}, s \in \mathcal{S}\}$ denotes computation allocation matrix.

Because the BSs can be gird-powered, we only consider the transmission energy consumption [14], [17]. So the total energy consumption of the remote computation can be written as

$$E_k^r(W) = E_k^t(W) = P_k T_k^t(W). \tag{8}$$

D. MDs QoE

Let $B = \{b_{k,s} \mid b_{k,s} \in \{0,1\}, k \in \mathcal{K}, s \in \mathcal{S}\}$ denote the offloading strategy matrix, which means not only whether to offload but also where to offload. $b_{k,s} = 1$ denotes that the MD k offloads the task to the MEC server s; otherwise, $b_{k,s} = 0$.

In a MEC system, the QoE is determined mainly by task completion time, i.e., T_k, and energy consumption, i.e.,E_k. Specifically, T_k and E_k can be expressed as

$$T_k(B,W,F) = \sum_{s\in\mathcal{S}} b_{k,s} \underbrace{T_k^r(W,F)}_{\text{the total remote computation completion time}} + (1-\sum_{s\in\mathcal{S}} b_{k,s}) \underbrace{T_k^l}_{\text{the local computation completion time}} . \tag{9}$$

$$E_k(B,W) = \sum_{s\in\mathcal{S}} b_{k,s} \underbrace{E_k^r(W)}_{\text{the total remote energy consumption}} + (1-\sum_{s\in\mathcal{S}} b_{k,s}) \underbrace{E_k^l}_{\text{the local energy computation}} . \tag{10}$$

When task k is completed locally (i.e., $b_{k,s} = 0$), we set $T_k^r(W,F) = 0$ and $E_k^r(W) = 0$. If $b_{k,s} = 1$, there are $T_k^l = 0$ and $E_k^l = 0$.

III. FUNDAMENTAL PROBLEM

In order to satisfy the QoE of MDs, we formulated the problem as the MDs' energy consumption minimization problem under the delay constraint, which is a MINLP problem. Furthermore, we transform the problem into a mixed integer non-linear convex problem which is easy to solve.

A. JOINT COMPUTATION OFFLOADING AND RESOURCE ALLOCATION PROBLEM

Lots of existing works have detailed studied "to offload or not", like [14], [15]. We simplified their researches as follows:

$$\begin{matrix} b_{k,0} = 1 : \{E_k^l < E_0\} \cap \{T_k^l < \tau_k\} \\ b_{k,0} = 0 : otherwise \end{matrix} , \quad \forall k \in \mathcal{K}.$$

Here, $b_{k,0} = 1$ denotes task k completed locally; otherwise, $b_{k,0} = 0$. We set E_0 as energy threshold and τ_k as the delay threshold to restrict the maximal cost. In this paper, we focus on the offloading strategy that is "offload to which one".

We only consider the task offloading MDs in this section, and the set $\mathcal{K}$ is updated as $\mathcal{K}'$, $\mathcal{K}' \subset \mathcal{K}$. Concerning with the energy computation, we formulate the joint optimization of computation offloading and resource allocation problem as follows:

$$\begin{aligned}
\mathcal{P} : \min_{B,W,F} \; & P(B,W,F) = \sum_{k\in\mathcal{K}'}\sum_{s\in\mathcal{S}} b_{k,s}E_k^r(W) \\
s.t.\; & C1 : b_{k,s} \in \{0,1\}, \quad \forall k \in \mathcal{K}', \forall s \in \mathcal{S} \\
& C2 : \sum_{s\in\mathcal{S}} b_{k,s} = 1, \quad \forall k \in \mathcal{K}' \\
& C3 : w_{k,n,s} \in \{0,1\}, \quad \forall k \in \mathcal{K}', \forall n \in \mathcal{N}, \forall s \in \mathcal{S} \\
& C4 : \sum_{k\in\mathcal{K}'}\sum_{s\in\mathcal{S}} w_{k,n,s} \le 1, \quad \forall n \in \mathcal{N} \\
& C5 : \sum_{n\in\mathcal{N}} w_{k,n,s} \le b_{k,s}N, \quad \forall k \in \mathcal{K}', \forall s \in \mathcal{S} \\
& C6 : 0 \le f_{k,s} \le F_s, \quad \forall k \in \mathcal{K}', \forall s \in \mathcal{S} \\
& C7 : f_{k,s} \le b_{k,s}F_s, \quad \forall k \in \mathcal{K}', \forall s \in \mathcal{S} \\
& C8 : \sum_{s\in\mathcal{S}} b_{k,s}T_k^r(W,F) \le \tau_k, \quad \forall k \in \mathcal{K}'
\end{aligned}$$

Constraint $C1$ states that $b_{k,s}$ is the offloading decision and it is the binary variable. Constraint $C2$ shows that each MD can only offload its task to one MEC server. According to $C3$ and $C4$, the $w_{k,n,s}$ is the binary variable of subcarrier allocation, and at each offloading decision, each subcarrier can be exclusively assigned to one MD. For any BS, $C5$ ensures that subcarrier assigned to any MD cannot exceed the maximum available subcarrier. $C6$ and $C7$ are the computation resource allocation constraints. $C6$ limits the range of and $F_{k,s}$, and $C7$ insures that the total resources assigned for one MEC server are less than the maximum instructions per second allowed at the MEC server s. The request of the total remote computation time on each offloaded task cannot exceed the hard deadline, thus we adopt constraint $C8$.

The above problem is a MINLP non-convex problem, which is NP-hard in general. Due to the mutual constraints of the three optimized variables, we can simplify the optimization problem through variable fusion.

B. PROBLEM TRANSFORMATION

According to the relationships among $b_{k,s}$, $w_{k,n,s}$ and $f_{k,s}$, we have the following constraints:

$$\begin{cases} b_{k,s} = 0 \Longleftrightarrow \sum_{n\in\mathcal{N}} w_{k,n,s} = 0, \quad \forall k \in \mathcal{K}', \quad \forall s \in \mathcal{S}. \\ b_{k,s} = 0 \Longleftrightarrow f_{k,s} = 0, \quad \forall k \in \mathcal{K}', \quad \forall s \in \mathcal{S}. \end{cases} \tag{11}$$

$b_{k,s}$ represents the decision variable. When the task k is not offloaded to the BS s, the BS will not be allocated computing resource to k. Corresponding, subcarriers will also not establish a connection between this task-server pair. Hence, we can combine the variable $b_{k,s}$ into $w_{k,n,s}$ and $f_{k,s}$, and details are as follows:

$$\begin{aligned}
&\sum_{s\in\mathcal{S}} b_{k,s}E_k^r(W) \\
&= \sum_{s\in\mathcal{S}} b_{k,s}P_k \frac{D_k}{B_N \sum_{n\in\mathcal{N}} w_{k,n,s}\log_2(1+\frac{g_{k,n,s}P_k}{\sigma^2})} \\
&= P_k \frac{D_k}{B_N \sum_{s\in\mathcal{S}}\sum_{n\in\mathcal{N}} w_{k,n,s}\log_2(1+\frac{g_{k,n,s}P_k}{\sigma^2})} \\
&= E_k^r(W)'.
\end{aligned} \tag{12}$$

$$\sum_{s\in\mathcal{S}} b_{k,s}T_k^r(W,F)$$

$$\begin{aligned}
&= \sum_{s\in\mathcal{S}} b_{k,s}\left(\frac{D_k}{B_N\sum\limits_{n\in\mathcal{N}} w_{k,n,s}\log_2(1+\frac{g_{k,n,s}P_k}{\sigma^2})} + \frac{D_kX_k}{f_{k,s}}\right)\\
&= \frac{D_k}{B_N\sum\limits_{s\in\mathcal{S}}\sum\limits_{n\in\mathcal{N}} w_{k,n,s}\log_2(1+\frac{g_{k,n,s}P_k}{\sigma^2})} + \frac{D_kX_k}{\sum\limits_{s\in\mathcal{S}} f_{k,s}}\\
&= T_k^r(W,F)'. \qquad (13)
\end{aligned}$$

With the Eq.(12) and Eq.(13), $\mathcal{P}$ can be equivalently transformed into $\mathcal{P}1$ as follows:

$$\begin{aligned}
\mathcal{P}1:\ &\min_{B,W,F} P1(B,W,F) = \sum_{k\in\mathcal{K}'} E_k^r(W)'\\
&s.t.\ C1-C7\\
&\qquad C9: T_k^r(W,F)' \le \tau_k, \quad \forall k\in\mathcal{K}'
\end{aligned}$$

The simplified problem $\mathcal{P}1$ is a mixed integer convex non-linear programming. Next, we present the BnB algorithm and combinational algorithm to get the solution.

IV. PROPOSED ALGORITHM

According to the above problem transformation, we first use BnB algorithm to get the global optimal solution of $\mathcal{P}1$. Then, we propose a combinational algorithm to obtain the suboptimal solution. That is, we obtain the optimal and suboptimal solution of the original problem.

A. OPTIMAL BRANCH-AND-BOUND ALGORITHM

First, we briefly describe the BnB algorithm used to solve the minimization problem. In general, a typical BnB algorithm consists of two parts: branching and bounding. All feasible solution spaces are repeatedly divided into smaller and smaller subsets, which is called branching. A lower bound of the target (for solving a minimum problem) is computed for the solution set in each subset, which is called a bounding.

After each branching, some branches can be effectively removed from the search tree by pruning. The premise of pruning are as follows [25]:

(1)The problem after branching is infeasible.

(2)The lowest bound of the problem after branching is larger than the best-known objective value.

In this way, many subsets may be left out of consideration. Based on the pruning conditions, we propose two methods to reduce the complexity of BnB algorithm.

Firstly, in order to better meet the first pruning condition, we add two linear redundant constraints that further constrain the problem and can be omitted without loss of generality. So the problem $\mathcal{P}1$ can be further written as:

$$\begin{aligned}
\mathcal{P}2:\ &\min_{B,W,F} P2(B,W,F) = \sum_{k\in\mathcal{K}'} E_k^r(W)'\\
&s.t.\ C1-C7, C9\\
&\qquad C10: b_{k,s} \le \sum_{n\in\mathcal{N}} w_{k,n,s}, \quad \forall k\in\mathcal{K}',\ \forall s\in\mathcal{S}\\
&\qquad C11: \sum_{s\in\mathcal{S}}\sum_{n\in\mathcal{N}} w_{k,n,s} \le \sum_{s\in\mathcal{S}} f_{k,s}N, \quad \forall k\in\mathcal{K}'
\end{aligned}$$

Algorithm 1 Optimal Branch-and-Bound Algorithm

```
Initialization: $\mathcal{I} = (o^{(0)}, \mathcal{L}_0^{(0)}, \mathcal{L}_1^{(0)})$, $o^* = +\infty$, $\mathcal{L}_0^{(0)} = \mathcal{L}_1^{(0)} = \emptyset$, and $\mathcal{L}$.
$n \leftarrow 0$
while $\mathcal{I} \neq \emptyset$ do
    Based on $\hat{o} = \min_{(o,\mathcal{L}_0,\mathcal{L}_1)\in\mathcal{I}} o$, choose $(\hat{o}, \hat{\mathcal{L}}_0, \hat{\mathcal{L}}_1)$ and update $\mathcal{I} = \mathcal{I} \setminus (\hat{o}, \hat{\mathcal{L}}_0, \hat{\mathcal{L}}_1)$.
    Choose one task-subcarrier-server node with the highest priority, i.e., $(k^*, n^*, s^*) = \arg\max_{(k,n,s)\in\mathcal{L}\setminus(\mathcal{L}_0\cup\mathcal{L}_1)} \hat{\xi}_p(k,n,s)$.
    Set $n = n + 1$.
    Update $\mathcal{L}_0^{(B1,n)} = \hat{\mathcal{L}}_0 \cup \{(k^*, n^*, s^*)\}$, $\mathcal{L}_1^{(B1,n)} = \hat{\mathcal{L}}_1$, $\mathcal{L}_0^{(B2,n)} = \hat{\mathcal{L}}_0$, and $\mathcal{L}_1^{(B2,n)} = \hat{\mathcal{L}}_1 \cup \{(k^*, n^*, s^*)\}$.
    Solve the sub-problems $(o^{(B_i,n)}, \mathcal{L}_0^{(B_i,n)}, \mathcal{L}_1^{(B_i,n)})^{CR}$, $i = 1, 2$. If no feasible solution, set $o^{(B_i,n)} = +\infty$.
    if $o^{(B_i,n)} < o^*$, $i = 1, 2$ then
        if $\mathcal{L} == \mathcal{L}_0^{(B_i,n)} \cup \mathcal{L}_1^{(B_i,n)}$ then
            Let $o^* = o^{(B_i,n)}$, $\mathcal{L}_0^* = \mathcal{L}_0^{(B_i,n)}$, $\mathcal{L}_1^* = \mathcal{L}_1^{(B_i,n)}$.
        else
            Update $\mathcal{I} = \mathcal{I} \cup \{(o^{(B_i,n)}, \mathcal{L}_0^{(B_i,n)}, \mathcal{L}_1^{(B_i,n)})\}$.
        end if
    end if
    Prune the branches.
end while
return $o^*$, $\mathcal{L}_0^*$, $\mathcal{L}_1^*$
```

For any k and s, $C10$ limits that when $b_{k,s}$ is 0, $w_{k,n,s}$ is also 0; and when $b_{k,s}$ is 1, $w_{k,n,s}$ is greater than or equal to 1. For any k, $C11$ shows that the number of subcarriers allocated to it must be less than or equal to the product of the computing resources allocated to it by the base station and the total number of subcarriers.

Then, carefully choosing the branch of search tree and the direction of search can better satisfy the second pruning condition, and can get the best-known value as soon as possible. This view will be described in detail in the following.

The BnB algorithm is presented in **Algorithm 1**. Define the set $A = B\bigcup W$, where $a_{k,n,s} \in \{0,1\}$ denote all of the binary variables instead of $b_{k,s}$ and $w_{k,n,s}$. Specially, let $n = 0$ (i.e., $\forall n \notin \mathcal{N}$) when $a_{k,n,s} \in B$. In this case, $a_{k,0,s}$ is a two-dimensional variable with respect to task-server pairs. We define the set of all the task-subcarrier-server pairs $\mathcal{L} = \{(k,n,s) \mid \forall k \in \mathcal{K}', \forall n \in \mathcal{N}, \forall s \in \mathcal{S}\}$ as the branch and bound nodes set. In addition, we extend two sets $\mathcal{L}_0 = \{(k,n,s) \mid a_{k,n,s} = 0, \forall k \in \mathcal{K}', \forall n \in \mathcal{N}, \forall s \in \mathcal{S}\}$ and $\mathcal{L}_1 = \{(k,n,s) \mid a_{k,n,s} = 1, \forall k \in \mathcal{K}', \forall n \in \mathcal{N}, \forall s \in \mathcal{S}\}$. Based on the $\mathcal{L}_0$ and $\mathcal{L}_1$, the problem $\mathcal{P}2$ can be equivalent to:

$$\begin{aligned}
\mathcal{P}3:\ &\min_{B,W,F} P3(B,W,F) = \sum_{k\in\mathcal{K}'} E_k^r(W)'\\
&s.t.\ C2, C4-C7, C9-C11\\
&\qquad C12: a_{k,n,s} = 0, \quad \forall (k,n,s) \in \mathcal{L}_0\\
&\qquad C13: a_{k,n,s} = 1, \quad \forall (k,n,s) \in \mathcal{L}_1
\end{aligned}$$

$$C14: a_{k,n,s} \in \{0,1\},\ (k,n,s) \in \mathcal{L} \setminus (\mathcal{L}_0 \cup \mathcal{L}_1)$$

When $\mathcal{L}_0$ and $\mathcal{L}_1$ are fixed, the continuous relaxation of problem $\mathcal{P}3$ can be formulated as:

$$\begin{aligned}\mathcal{P}4: &\min_{B,W,F} P4(B,W,F) = \sum_{k\in\mathcal{K}'} E_k^r(W)' \\ &s.t.\ C2, C4-C7, C9-C13 \\ &\quad C15: a_{k,n,s} \in [0,1], (k,n,s) \in \mathcal{L} \setminus (\mathcal{L}_0 \cup \mathcal{L}_1)\end{aligned}$$

Obviously, the optimal value of problem $\mathcal{P}4$ is the lower bound for problem $\mathcal{P}3$. We represent $\mathcal{P}3$ and $\mathcal{P}4$ as related parameter tuples $(o, \mathcal{L}_0, \mathcal{L}_1)$ and $(o, \mathcal{L}_0, \mathcal{L}_1)^{CR}$, respectively. o is the optimal objective value of $\mathcal{P}4$, which is the lower bound of $\mathcal{P}3$. Then, we define $\mathcal{I}$ as the set of brunch problem and o^* as the best-known objective value. The main processes of the BnB algorithm are as follows:

1) BRANCHING

In each branching iteration, the problem that attains the minimum lower bound, denoted as $(\hat{o}, \hat{\mathcal{L}_0}, \hat{\mathcal{L}_1})$, is chosen to branch. Then, we select the task-subcarrier-server node with the highest priority (k^*, n^*, s^*), and the problem is divided into two smaller problems by setting the integer variable a_{k^*,n^*,s^*} to a binary value of 0 or 1. It is apparent that the priority function is important to reduce the complexity of BnB algorithm, because many problems may be pruned if we get the best-known objective value. So, the priority function for problem $(\hat{o}, \hat{\mathcal{L}_0}, \hat{\mathcal{L}_1})$ is defined as $\hat{\xi}_p(k,n,s) = \frac{g_{k,n,s} D_k}{d_{k,s}}$, where $d_{k,s}$ represents the distance between MD k and server s. In addition, when the node is a member of set B, we set $g_{k,n,s} = 1$ [26], [27].

2) BOUNDING AND PRUNING

Based on the selected branch, we count the lower bound of sub-problems $(o^{(B1,n)}, \mathcal{L}_0^{(B1,n)}, \mathcal{L}_1^{(B1,n)})^{CR}$ and $(o^{(B2,n)}, \mathcal{L}_0^{(B2,n)}, \mathcal{L}_1^{(B2,n)})^{CR}$. According to [28], global convergence is guaranteed by simple bound. Compare the new solutions $(o^{(B1,n)}, o^{(B2,n)})$ with the current best-known objective value, and update the o^* with the small one. The problem whose lower bound is smaller than o^* will be put into $\mathcal{I}$. Otherwise, it will be pruned.

B. SUBOPTIMAL INTELLIGENT HEURISTIC ALGORITHM

Although **Algorithm 1** can obtain the global optimal solution, the convergence speed is still a problem that cannot be ignored for large networks. In order to facilitate practical application, we propose a fast and intelligent heuristic greedy algorithm to find the suboptimal solution of problem $\mathcal{P}$.

As shown in **Algorithm 2**, we break down the problem into the minimized energy consumption problem with different MD. Using the greedy principle, the local optimal solution for each sub-problem is obtained. Finally, according to the local optimal solution of each sub-problem, the final solution of the problem is obtained by accumulating.

Task Ψ_k with the largest input-data size in all unuploaded tasks set is first considered. At this time, server s^* which has the nearest distance to this task has the highest priority to serve it. When server s^* has enough resources to support task k^* to be completed within the maximum accomplished deadline τ_k, this server will allocate resources f_{k^*,s^*} to this task according to constraint $C8$, i.e., $f_{k^*,s^*} = \frac{D_{k^*} X_{k^*}}{\tau_{k^*} - \frac{D_{k^*}}{R_{k^*,s^*}}}$. Otherwise, task k^* continues to look for the appropriate server. In terms of quantity, subcarriers are allocated to each task as equitably as possible. To make the system perform better, the task with larger input-data size will be allocated to more subcarriers. Under the condition of the number mentioned above, subcarriers with large channel gain are preferentially selected. It is worth noting that the suboptimal heuristic algorithm cannot always find the solution to the problem. However, it has very satisfactory performance with sufficient resources.

Algorithm 2 Suboptimal Intelligent Heuristic Algorithm

```
1:  Initialize: $\mathcal{S}' = \mathcal{S}$. Select task $\Psi_{k^*} = \arg\max_{k\in\mathcal{K}'} D_k$.
2:  if Subcarriers can be equally distributed to each MD then
3:      The number of subcarriers of task $\Psi_{k^*}$ is $J = N/K$, where $N$ denotes the number of current subcarriers and $K$ denotes the number of current MDs.
4:  else
5:      $J = [N/K] + 1$.
6:  end if
7:  Update $\mathcal{K}' \triangleq \mathcal{K}' \setminus \{k^*\}$.
8:  if $\mathcal{S}' \neq \emptyset$ then
9:      Choose server $s^* = \arg\min_{s\in\mathcal{S}'} d_{k^*,s}$ for task $\Psi_{k^*}$.
10:     Update $\mathcal{S}' = \mathcal{S}' \setminus \{s^*\}$.
11:     if $f_{k^*,s^*} \leq F_{s^*}$ then
12:         for $j \in J$ do
13:             Subcarrier $n_j^* = \arg\max_{n\in\mathcal{N}} g_{k^*,n,s^*}$, and update $\mathcal{N} = \mathcal{N} \setminus \{n_j^*\}$.
14:         end for
15:         Update $F_{s^*} = F_{s^*} - f_{k^*,s^*}$.
16:         Go to step 1.
17:     else
18:         Go to step 8.
19:     end if
20: else
21:     Suboptimal heuristic algorithm fails.
22: end if
```

C. COMBINATIONAL ALGORITHM

Since **Algorithm 2** has the probability of failure, we propose the combinational algorithm as shown in **Algorithm 3**. **Algorithm 2** is first used to find the suboptimal solution. If the feasible solution cannot be found, **Algorithm 1** will be adopted.

V. PERFORMANCE EVALUATIONS

In this part, we evaluate the numerical results to demonstrate the performance of our proposed algorithms and compare

Algorithm 3 Combinational Joint Computation Offloading and Resource Allocation (CJCORA)

```
Algorithm 2 is first used.
if There is no feasible solution via Algorithm 2 then
    Algorithm 1 is adopted.
else
    Return the feasible solution of Algorithm 2.
end if
```

TABLE 1. Simulation parameters.

MEC System Parameters	Values
Subcarrier bandwidth B_N	12.5 kHz
Background noise σ^2	$10^{(-13)}$ W
Maximum transmission power P^m	600 mW
Input-data size D_k	1000-1500 bits
The computation workload/intensity X_k	1000 cycles/bit
Maximum accomplished deadline τ_k	9-10 ms
The CPU frequency of MDs F_k^l	0.6-0.7 GHz
The CPU frequency of MEC servers F_s	2.4-2.5 GHz

them to conventional schemes. MDs and MEC servers are distributed uniformly and independently in a circle area with a radius of 100 meters. The uplink channel gains are generated using a distance-dependent path loss function is expressed as $PL = 128.1 + 37.6 \log_{10}(d_{k,s})$ where d in km, and the small scale fading adopts the Rayleigh fading model [29] [30]. Unless other wise stated, we consider $K = 20$, $S = 7$ and $N = 128$. Other parameters are in Table 1.

We compare the energy consumption performance of our proposed algorithm against the following algorithms:

1): **Local computation algorithm (LCA).** It means that there is no offloading. All tasks are performed locally.

2): **Random offloading and joint resource allocation (ROJRA).** Each MD randomly selects one MEC server to complete task, and subcarriers and computing resources are allocated jointly.

3):**Greedy offloading and average resource allocation (GOARA).** MDs greedily choose the closer server to offload, and the servers' computing resources are equally allocated to the MDs connected to it. Subcarriers are distributed as evenly as possible and the redundant ones are given to the task with larger input-data size. The selection of each subcarrier between MDs and servers is also from the perspective of fairness, and the subcarriers with the same channel gain are selected as far as possible.

A. SUBOPTIMALITY AND TIME COMPLEXITY BEHAVIOR OF Algorithm 1 AND Algorithm 3

Firstly, in order to prove the suboptimality of our proposed combinational algorithm, we compare its performance with our proposed **Algorithm 1**, which can find the optimal solution of the problem. Because the high complexity of **Algorithm 1**, we carry out simulation in a small network setting. Fig.2 and TABLE. 2 are obtained by using parameters $S = 2$ and $N = 16$ through 200 independent

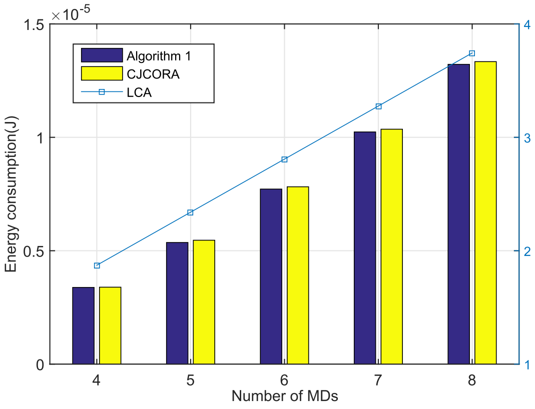

FIGURE 2. Comparison of Algorithm 1 and CJCORA algorithm: Energy consumption of MDs.

TABLE 2. Average algorithm time.

Number of MDs	Runtime of Algorithm 1 (s)	Runtime of CJCORA (ms)
4	133	16 ± 0.7
5	167	16 ± 0.7
6	203	16 ± 0.7
7	270	16 ± 0.7
8	337	16 ± 0.7

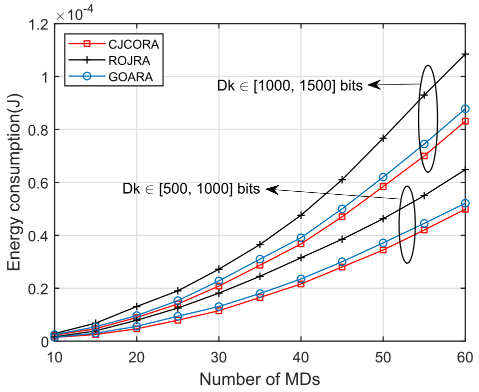

FIGURE 3. Comparison of energy consumption against different number of MDs, evaluated on two different task input-data size D_k.

realizations. It can be seen that the energy consumption of the proposed CJCORA algorithm is very close to that of the optimal BnB algorithm, but the average running time of CJCORA algorithm is far less than that of the BnB algorithm. Moreover, the average running time of **Algorithm 1** is greatly affected by the changes of MDs in a small range, while the running time of **Algorithm 3** is not affected. It is shown that the CJCORA algorithm is more beneficial to the actual deployment with a little performance loss. We can also observe that no matter which method we proposed is adopted, the MD will save 10000× energy consumption compared to LCA.

B. EFFECT OF MDs AND TASKS

Next, Fig. 3 presents the performance of three different algorithms versus the number of MDs wishing to offload

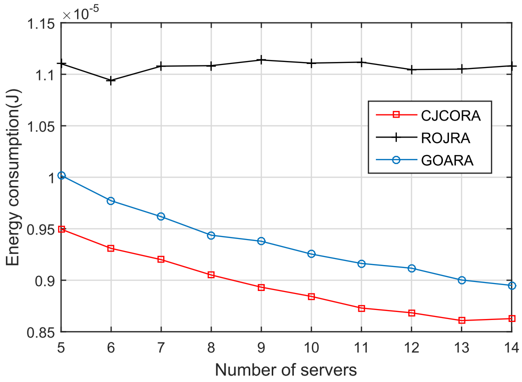

FIGURE 4. The performance of CJCORA, ROJRA and GOARA versus the number of servers.

their tasks. Particularly, the number of MDs per unit ranges from 1 to 10, and we compare them in two cases with different task input-data sizes. Note that we set the number of servers $S = 7$ and the number of subcarriers $N = 128$. It's not hard to see from Fig. 3 that our proposed CJCORA algorithm performs better than others. This is due to the increase in the number of MDs, and the corresponding number of tasks also increases, resulting in a large increase in the total energy consumption of MDs in the system. What's more, it can be seen that the size of the task input data has a positive impact on the system energy consumption. Compared with the CJCORA algorithm, the energy consumption of the ROJRA algorithm is much different from that of the CJCORA algorithm, while the GOARA algorithm has less difference. This shows that the factor that has a greater impact on system energy consumption is the offload allocation.

C. EFFECT OF SERVERS

As seen from Fig. 4, our proposed CJCORA algorithm always maintains the best performance. Moreover, the energy consumption of the GOARA algorithm and the CJCORA algorithm decreases as the number of servers increases, while the ROJRA algorithm is not greatly affected by the number of servers. This is because the CJCORA algorithm and the GOARA algorithm are sensitive to the offload allocation. When the number of servers increases, the MDs can select more servers, and they have a greater chance of selecting servers that are more beneficial to themselves, thereby reducing system energy consumption. The ROJRA algorithm is more sensitive to resource allocation, and the change in the number of servers does not affect its choice of the server.

VI. CONCLUSION

In this paper, we discuss the joint computation offloading and resources optimization in a multi-access MEC system. In the resource-constrained system, we represent the optimization problem as a non-convex MINLP problem with three mutually constrained variables, which is difficult to solve. Through the combination of variables, the problem is transformed into a mixed integer non-linear convex problem. Then, we propose the bound improving branch-and bound (BnB) algorithm and combinational joint computation offloading and resource allocation (CJCORA) algorithm to obtain the optimal solution and the suboptimal solution respectively. The simulation results show that the performance of the proposed CJCORA algorithm is very close to the optimal BnB algorithm, and its performance is significantly improved compared to other algorithms.

In addition, future work is to consider a more optimized algorithm to get the optimal solution, which is a more efficient runtime BnB algorithm. Since the problem we have developed is a MINLP problem, the MINLP problem is usually NP-hard and it is difficult to obtain an optimal solution. The traditional methods of solving these MINLP problems are based on mathematical optimization techniques that only get suboptimal solutions and often have a prohibitive complexity of real-time implementation. Nowadays, in the direction of deep learning, wireless networks are becoming smarter [31], [32]. There have been articles that study the branching strategy [33] and pruning strategy [34] to efficiently accelerate the most time-consuming branching process of the BnB algorithm. Applying the BnB algorithm under deep learning to this scenario is the focus of our next step.

REFERENCES

[1] W. Xia, T. Q. S. Quek, J. Zhang, S. Jin, and H. Zhu, "Programmable hierarchical C-RAN: From task scheduling to resource allocation," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 2003–2016, Mar. 2019.

[2] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu, "Unsupervised learning-based fast beamforming design for downlink MIMO," *IEEE Access*, vol. 7, pp. 7599–7605, Jan. 2019.

[3] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[4] X. Wei, S. Wang, A. Zhou, J. Xu, S. Su, S. Kumar, and F. Yang, "MVR: An architecture for computation offloading in mobile edge computing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 232–235.

[5] B. Dab, N. Aitsaadi, and R. Langar, "A novel joint offloading and resource allocation scheme for mobile edge computing," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–2.

[6] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.

[7] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell mobile edge computing," in *Proc. Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–5.

[8] 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.

[9] Y. Yu, J. Zhang, and K. B. Letaief, "Joint subcarrier and CPU time allocation for mobile edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[10] M. Liu, T. Song, J. Hu, J. Yang, and G. Gui, "Deep learning-inspired message passing algorithm for efficient resource allocation in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 641–653, Jan. 2019.

[11] M. Liu, T. Song, and G. Gui, "Deep cognitive perspective: Resource allocation for NOMA-based heterogeneous IoT with imperfect SIC," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2885–2894, Apr. 2019.

[12] W. Xia, J. Zhang, T. Q. S. Quek, S. Jin, and H. Zhu, "Energy-efficient task scheduling and resource allocation in downlink C-RAN," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[13] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and doa estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.

[14] 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.

[15] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, 2017.

[16] K. Cheng, Y. Teng, W. Sun, A. Liu, and X. Wang, "Energy-efficient joint offloading and wireless resource allocation strategy in multi-MEC server systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[17] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "Energy efficient computation offloading for multi-access mec enabled small cell networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[18] 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.

[19] C. Wang, C. Dong, J. Qin, X. Yang, and W. Wen, "Energy-efficient offloading policy for resource allocation in distributed mobile edge computing," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 366–372.

[20] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[21] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12313–12325, Dec. 2018.

[22] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing," *IEEE Access*, vol. 6, pp. 19324–19337, 2018.

[23] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[24] 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.

[25] Z. Yu, K. Wang, H. Ji, and V. C. M. Leung, "Joint multiuser admission control and downlink beamforming for green cloud-RANs via semidefinite relaxation," in *Proc. 19th Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, Nov. 2016, pp. 244–249.

[26] T. D. Hoang, L. B. Le, and T. Le-Ngoc, "Energy-efficient resource allocation for D2D communications in cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sep. 2015, pp. 2251–2256.

[27] W. Xia, J. Zhang, T. Q. S. Quek, S. Jin, and H. Zhu, "Power minimization-based joint task scheduling and resource allocation in downlink C-RAN," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7268–7280, Nov. 2018.

[28] H. Tuy, F. Al-Khayyal, and P. T. Thach, "Monotonic optimization: Branch and cut methods," in *Essays and Surveys in Global Optimization*. Boston, MA, USA: Springer, 2005, pp. 39–78.

[29] X. Chu, D. López-Pérez, Y. Yang, and F. Gunnarsson, *Heterogeneous Cellular Networks: Theory, Simulation and Deployment*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[30] P.-F. Cui, W.-J. Lu, Y. Yu, B. Xue, and H.-B. Zhu, "Off-body spatial diversity reception using circular and linear polarization: Measurement and modeling," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 209–212, Jan. 2018.

[31] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *CoRR*, Dec. 2018. [Online]. Available: http://arxiv.org/abs/1812.02858

[32] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.

[33] M. Lee, G. Yu, and G. Y. Li, "Learning to branch: Accelerating resource allocation in wireless networks," *CoRR*, Mar. 2019. [Online]. Available: http://arxiv.org/abs/1903.01819

[34] H. He, H. Daume, III, and J. Eisner, "Learning to search in branch and bound algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3293–3301.

XIAOTONG YANG received the B.S. degree in communication engineering from Shandong Normal University (SDNU), in 2017. She is currently pursuing the master's degree with the Nanjing University of Posts and Telecommunications (NUPT), where she involved in mobile edge computing.

XUEYONG YU was born in Jiangxi, China, in 1979. He received the Ph.D. degree in electromagnetic field and microwave technology from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2016, where he is currently an Associate Professor. His current research interests include the Internet of Thing (IoT), mobile edge computing, and radio resource management on heterogeneous wireless networks.

HAO HUANG (S'18) received the B.S. degree in photoelectric information science and engineering from the Nanjing University of Posts and Telecommunications (NUPT), Nanjing, China, in 2017, where he is currently pursuing the Ph.D. degree. His research interest includes deep learning and its application in wireless communications.

HONGBO ZHU received the B.S. degree in communications engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 1982, and the Ph.D. degree in information and communications engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 1996. He is currently a Professor with the Nanjing University of Posts and Telecommunications, Nanjing. He is also the Head of the Coordination Innovative Center of IoT Technology and Application, which is the first governmental authorized Coordination Innovative Center of IoT in China. He also serves as a referee or an expert in multiple national organizations and committees. He has authored or coauthored over 200 technical papers published in various journals and conferences. He is currently leading a big group and multiple funds on the IoT and wireless communications with current focus on architecture and enabling technologies for the Internet of Things. His research interests include mobile communications, wireless communication theory, and electromagnetic compatibility.