IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.DOI

Energy-efficient computation offloading and resource allocation in SWIPT-based MEC Networks

XUEFEI E, ZHONGGUI MA, KAIHANG YU

School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China (e-mail: aoxuefei97@163.com, zhongguima@ustb.edu.cn, 17801069926@163.com)

Corresponding author: Zhonggui Ma (e-mail: zhongguima@ustb.edu.cn).

This work is supported by the National Natural Science Foundation of China (61822104, 61872221) and the the Fundamental Research Funds for the Central Universities(FRF-DF-20-12, FRF-GF-18-017B).

ABSTRACT Recent years, 5G networks have become an important role in accelerating the development of social intelligence. But it also increases energy consumption and data flow. In order to guarantee the experience of network users, an adaptive SWIPT-based MEC network is proposed. The network consists of multiple user equipment (UE) and multiple mobile edge computing (MEC) servers. The MEC server can make up for the shortcomings of the UE's insufficient computing capability, and Simultaneous Wireless Information and Power Transfer (SWIPT) can send energy to the UE without pollution to make up for the shortcomings of limited battery energy. Added, we increase the utilization of sub-channels, improve the adaptability of the SWIPT-based MEC network to the environment, lengthen the battery life, and optimize the UE's energy efficiency of the network. We also propose a three-part alternative optimization algorithm framework based on the categories of optimization variables. The first part combines the Alternating Direction Multiplier Method (ADMM) and Dinkelbach's algorithm to optimize continuous variables. And for directly optimize the integer variables of the other two parts, without converting them to continuous variables. The second part adjusts the offloading decision by comparing the energy consumption of the two computation modes, and the third part proposes the integer Bat algorithm to assign sub-channel. The simulation results show that the energy efficiency of the binary offloading algorithm proposed in this paper is 11% higher than the approximate algorithm based on the binary offloading KKT algorithm, and the relationship between various preset parameters and energy efficiency in the network is discussed.

INDEX TERMS MEC, SWIPT, ADMM, Integer Bat algorithm, Dinkelbach

I. INTRODUCTION

THE bandwidth, capacity and data rate of current 4G mobile communications are seriously inadequate, and it is incompetent to stimulate the development of society and technology [1]. By 2023, there will be 8.7 billion handheld or personal mobile communication devices connected to the network [2]. Such a contradiction accelerated the birth of 5G networks. Compared with the previous 4G networks, 5G can increase the data rate to 10Gbps [3]. For today's society, 5G has become the driving force of development and has been used in many fields. Including Drone-assisted Networks [4], [5], Autonomous Driving Vehicles [6], Smart Communities [7], Internet-of-Vehicles (IoV) [8]. The widespread application of 5G makes devices toward smarter and more portable. Indeed, a great amount of data calculation can make the device smarter. But portable devices with low computing power and energy limited are obviously incapable to meet this case.

Currently, one of the best methods to solve the above problems is computational offloading [9]. By transmitting part or all of the calculation tasks to the terminal processor, UEs can reduce the time of calculations. In addition, A. Rudenko, et al. show that computational offloading can save energy significantly. It is worth mentioning that portable devices can reduce energy consumption by up to 51% when performing larger-scale computing tasks [10]. Therefore, Mobile Edge Computing (MEC) [11] has become a research hotspot. Compared with cloud computing, the MEC server is closer to UEs, which can reduce the consumption of transmission energy for the UEs [12]. MEC is also outstanding in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3047690, IEEE Access

improving QoS. With the high computing performance of the MEC server, calculation delay [13] and energy consumption [14] of UEs can be significantly decreased.

In the MEC network, part of UEs have limited battery capacity and cannot work for a long time without power supply [15]. One way to solve this problem is to increase the battery capacity. However, the development of battery technology is too slow to keep up with the demands of current UEs [16]. Therefore, wireless Energy Harvesting (EH) has attracted much attention in recent years. The concept of Simultaneous Wireless Information and Power Transfer (SWIPT) was proposed in 2008 [17], making it possible to transmit energy and information simultaneously. EH equipments of SWIPT has two modes: Time Switching (TS) and Power Switching (PS) [18]. EH and Information Decoding (ID) of TS mode is in different time slots. PS model divides the received signal into two signals with different powers and performs EH and ID at the same time. These two receiving modes enable SWIPT to cope with various MEC networks leisurely. Therefore, SWIPT-based MEC has become a key technology to overcome the limitation of battery performance [19].

More and more scholars choose to embed SWIPT into the MEC network. In [20], for improving the fairness of UEs, the authors use PS model to adjust the downlink transmission rate by changing the ratio of EH and ID. In [21], under the premise of each User Equipment (UE) has sufficient energy, the authors tend to increase the success rate of offloading as much as possible, and use stochastic geometry to asymptoticly analyze the model. Besides, Full-Duplex (FD) technology can receive and transmit signals at the same time, which give many opportunities to SWIPT-based MEC network and attracts a large number of scholars [22]–[24]. In [22], a SWIPT framework is proposed to combine FD with SWIPT, which can transmit energy in both uplink and downlink. The results show that the multi-user situation is optimal under high power conditions. This framework makes UE no longer worry about energy. A MEC framework is proposed in [23], which uses an iterative algorithm to minimize the uplink energy consumption in partial offload mode. In [24], the SWIPT-MEC network is equipped with Multiple-Input Multiple-Output (MIMO) and FD technologies, and jointly optimize the uplink and downlink to minimize energy consumption. OFDM can optimize the power and rate of each sub-carrier. But it is worse than Orthogonal Frequency-Division Multiplexing Access (OFDMA) in adapting to the time-varying environment [25]. Therefore, many scholars have recently used OFDMA technology to improve the capability of uplink. In [26], maximizing the throughput of the uplink in the OFDMA-SWIPT system is regarded as the research goal, and the cases of frequency division multiplexing (FDD) and time division multiplexing (TDD) are analyzed separately. M Li et al. proposed an OFDMA-based MEC system and developed a heuristic algorithm that calculates sub-carrier allocation and power allocation separately [27].

To the best of our knowledge, the introduction of SWIPT does improve the energy efficiency of the MEC network, but the channel allocation strategy in most of existed jobs is static. Besides, when the channel noise and interference lasts for a long time, UE will exhaust energy after a long local computation. That is fatal for people equipped with human embedded devices. To solve these problem, a multi-user and multi-server SWIPT-based MEC network is designed in this paper, which can provide uninterrupted power supply to all UEs. And added OFDMA to improve usability and stability. To simplify the algorithm, we divide the optimization variables into three categories by judging continuity and analyzing dependencies. And use the three-party alternate optimization algorithm framework to optimize them. This framework aims to maximize network energy efficiency and ensure that UEs have sufficient energy. The main contributions of this paper are listed as follows:

- Considering the dynamic channel allocation strategy, we propose a SWIPT-based MEC network with multiple UEs and multiple MEC servers, which ensures that each UE continuously receives energy and is only served by one MEC server. Besides, the computing ability of MEC servers and UEs in the SWIPT-based MEC network is limited.
- An energy efficiency optimization model of SWIPT-based MEC network is given and the power allocation, local computation intensity, offloading decision-making and sub-channel allocation are jointly optimized.
- To solve the energy efficiency optimization model of Mixed Integer Non-Linear Programming (MINLP), we divide the model into three parts according to the different variables, and jointly optimize them through three-part alternate optimization algorithm framework. In first part, for local computing frequency and uplink transmission power, we use Dinkelbach algorithm to transform the fractional objective functions into polynomial form. To accelerate the convergence, we use the ADMM algorithm to alternately optimize the sub-models, and the KKT condition is used as an optimization tool to optimize the alternate two branchs respectively. Second, the offloading strategy is determined by comparing the energy consumption of the two computation modes. Third, the integer bat algorithm is used for sub-channel allocation.
- The simulation results show that the energy efficiency of our proposed binary offloading algorithm is 11% higher than one of the approximate algorithm based on binary offloading KKT algorithm. Moreover, the increase of parameters such as delay, sub-channel bandwidth and number of sub-channels has a positive impact on energy efficiency. The results also show that equipping EH sub-channels for each UE will improve the energy efficiency of the SWIPT-based MEC network.

The rest of the paper are structured as follows. In Section II, the system model and energy efficiency optimization model of the SWIPT-based MEC network are given.

In Section III, The model is mathematically deformed and disassembled to simplify the solution of the problem. In Section IV, the solution algorithms are proposed for three sub-models. Simulation results and analysis are presented in Section V. Finally, Section VI makes a summary of the paper.

II. SYSTEM MODEL

The network in this paper is a subnet of the kubernetes network, mainly used for user clusters. When each UE enters the service range, its hardware information will be uploaded to etcd through the MEC server. All MEC servers can access etcd to obtain the required information and return the optimized local computing intensity to the served UEs. This paper only needs very little hardware information, so the energy consumption of uploading hardware information and return time can be ignored. As shown in Fig. 1, there are multiple single-antenna UEs and multiple MEC servers in the SWIPT-based MEC network. Let the set of MEC servers is denoted by $\mathcal{K} = \{1, 2, ..., K\}$ and the set of UEs is denoted by $\mathcal{I} = \{1, 2, ..., I\}$. The MEC servers based on binary offloading transmit information and energy to the UEs, and each MEC server is equipped with MIMO antennas, powerful computing chips and charging equipments. In addition, OFDMA is used in the uplink in order to decrease interference among sub-channels, save energy consumption and enhance the network's ability to respond to changes. The set of sub-channels is denoted by $\mathcal{N} = \{1, 2, ..., N\}$ and the sub-channels used by each UE cannot be shared. i,e,.

$$C3 : \sum_{i \in \mathcal{I}} w_{i,n} \leq 1 \quad \forall n \in \mathcal{N}$$

Here $W = \{W_{i,n} \mid w_{i,n} \in \{0, 1\}, \forall i \in \mathcal{I}, \forall n \in \mathcal{N}\}$ means the sub-channel n allocated to the UE_i. In this scenario, the computing capabilities of the $\mathcal{K}$ MEC servers are limited and only serve UEs within a fixed range. They need to deal with the computing tasks uploaded by the UEs and then return the results and energy through the $\mathcal{N}$ sub-channels. We assume that the calculation tasks for each UE are served by only one MEC. Therefore we have

$$C4 : \sum_{k \in \mathcal{K}} b_{i,k} \leq 1 \quad \forall i \in \mathcal{I}$$

$$C5 : 0 \leq \sum_{i \in \mathcal{I}} F_{i,k}^{\mathrm{UC}} \leq F_k \quad \forall k \in \mathcal{K}$$

where $F^{\mathrm{UC}} = \{F_{i,k}^{\mathrm{UC}} \mid \forall i \in \mathcal{I}, \forall k \in \mathcal{K}\}$ represents the frequency that the MEC_k spends in calculating the tasks of the UE_i. Let $B = \{b_{i,k} \mid b_{i,k} \in \{0, 1\}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}\}$ represents UE's offloading strategy. $b_{i,k} = 1$ represents that the UE_i offloads the computation task to the MEC_k; otherwise, $b_{i,k} = 0$.

A. ENERGY HARVESTING

In order to ensure sufficient energy for each UE in the MEC, we assume that the UE can continuously receive energy from the MEC. Besides, noise and interference is bad for

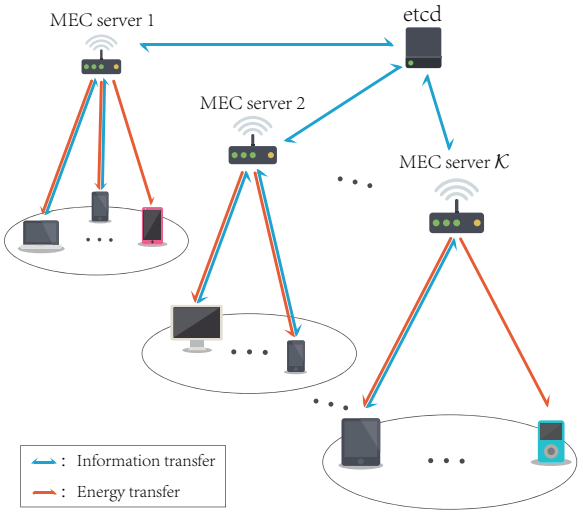

FIGURE 1. Multi-UE and Multi-server SWIPT-based MEC network.

ID, but beneficial for EH. Therefore, it is reasonable and valid to transmit energy to the local computing UE. Let $h_{i,n}$ represents the gain of the sub-channel n with channel estimation error allocated to the UE_i and σ^2 represents the white Gaussian noise power. Farther, we assume that the channel gain remains constant over a period of time [28]. Consider that computing is not the only energy-consuming behavior, so the received energy of the UE must be more than the calculated energy consumption. To ensure that the power of the UE will not be exhausted, this paper uses the maximum downlink power to transmit energy to the UE. According to [29], the maximum energy received by the UE_i in the uplink and downlink can be denoted as

$$E_i^{\mathrm{EH}} = \zeta_i \left(\sum_{k \in \mathcal{K}} P_{i,k}^{\mathrm{DL}} t_i^{\mathrm{tolerant}} h_{i,n,k}^{\mathrm{EH}} + \sigma^2 \right) \tag{1}$$

Where $\zeta = \{\zeta_1, \zeta_2, ..., \zeta_i, ..., \zeta_I\}$ denotes the energy absorption rate of all UEs. $P^{\mathrm{DL}} = \left\{ P_{i,k}^{\mathrm{DL}} \mid \forall i \in \mathcal{I}; \forall k \in \mathcal{K} \right\}$ represents the downlink transmission power between MEC_k and UE_i. And $h_{i,n,k}^{\mathrm{EH}}$ represents the gain of the sub-channel used for energy transfer between UE_i and MEC_k. These channels are fixed since each UE continuously receives energy. Finally, $t_i^{tolerant}$ indicates the latest time when the UE_i receives the calculation result. It is assumed that the UE_i receives signals from different subcarriers through a polyphase bandpass filter [30].

B. LOCAL COMPUTING

If the content of the calculation task is ignored, it can be expressed as D_i (bit). For a locally calculated UE_i, the CPU calculation ability is written as X_i (in CPU cycle per bit).

When UE_i selects the local computing mode, the total computation time can be expressed as

$$T_i^{\mathrm{LC}} = \frac{D_i X_i}{F_i^{\mathrm{LC}}} \tag{2}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3047690, IEEE Access

where F_i^{LC} represents the clock frequency of UE_i's CPU. For more practical, we set the upper limit of the calculation frequency for the CPU, denoted as F_i, which can be equivalently transformed into

$$C6: 0 \leq F_i^{\mathrm{LC}} \leq F_i \quad \forall i \in \mathcal{I}$$

According to [31], the energy consumption of each CPU cycle is proportional to the square of F_i^{LC} during the local computing, and it can be written as

$$p_i^{\mathrm{LC}} = k_0 \left(F_i^{\mathrm{LC}}\right)^2 \quad [k_0 = 1 \times 10^{-24}] \tag{3}$$

where k_0 is a constant coefficient related to the CPU of UE_i. Therefore, the energy consumed by the UE_i to calculate a complete task can be written as

$$E_i^{\mathrm{LC}} = \left(1 - \sum_{k \in \mathcal{K}} b_{i,k}\right) p_i^{\mathrm{LC}} D_i X_i \tag{4}$$

Here, the physical meaning of $\left(1 - \sum_{k \in \mathcal{K}} b_{i,k}\right)$ is to determine whether UE_i selects local computing.

C. COMPUTATION OFFLOADING

Each UE that chooses to computation offloading has two or more sub-channels, one of which is used to transmit energy. We assume the channel bandwidth of each sub-channel as B_N. So the number of sub-channels used by the UE_i to transmit computing tasks can be defined as $\left(\sum_{n \in \mathcal{N}} w_{i,n} - 1\right)$. $R_{i,n}^{\mathrm{UL}}$ represents the transmission rate of the sub-channel n, and it is denoted as

$$R_{i,n}^{\mathrm{UL}} = B_N log_2 \left(1 + \frac{\sum_{n \in \mathcal{K}} b_{i,k} h_{i,n,k} P_{i,n}^{\mathrm{UL}}}{\sigma^2}\right) \tag{5}$$

where $P^{\mathrm{UL}} = \left\{P_{i,n}^{\mathrm{UL}} \mid \forall i \in \mathcal{I}, \forall n \in \mathcal{N}\right\}$ represents the transmission power set of all UEs when they upload a computation task, and the maximum uplink power of each UE is expressed as P_i^{Max}. Then, the constraint can be expressed as

$$C7: \sum_{n \in \mathcal{N}} P_{i,n}^{\mathrm{UL}} \leq P_i^{\mathrm{Max}} \quad \forall i \in \mathcal{I}$$

The main steps of the computation offloading process include: MEC_k receives the computation task from the UE_i, then calculates the task, and returns the computation result to the UE_i. Since the length of the computation result is much shorter than the computation task, the time for returning the computation result can be negligible. On the premise of $R_{i,n}^{\mathrm{UL}} \neq 0$, the time for the MEC_k to receive the computing task of the UE_i can be expressed as T_i^{UL}, and the time of the MEC_k calculation task is T_i^{UC}. They are respectively given by

$$T_i^{\mathrm{UL}} = \frac{D_i}{\sum_{n \in \mathcal{N}} (w_{i,n} - \frac{1}{N}) R_{i,n}^{\mathrm{UL}}} \tag{6}$$

$$T_i^{\mathrm{UC}} = \frac{D_i X_i}{\sum_{k \in \mathcal{K}} F_{i,k}^{\mathrm{UC}}} \tag{7}$$

To simplify the model, we assume $t_i^{tolerant}$ is the maximum computation time for both local computing and computation offloading, which can be expressed as follow

$$C8: T_i^{\mathrm{LC}} \leq t_i^{\mathrm{tolerant}} \quad \forall i \in \mathcal{I}$$

$$C9: T_i^{\mathrm{UC}} + T_i^{\mathrm{UL}} \leq t_i^{\mathrm{tolerant}} \quad \forall i \in \mathcal{I}$$

Uplink energy consumption can be written as

$$E_i^{\mathrm{UL}} = \sum_{n \in \mathcal{N}} P_{i,n}^{\mathrm{UL}} T_i^{\mathrm{UL}} \tag{8}$$

D. ENERGY EFFICIENCY OPTIMIZATION MODEL OF SWIPT-BASED MEC NETWORK

For all UEs, the energy consumption must be less than the maximum energy gain E_i^{EH}. The corresponding constraint is written as

$$C10: E_i^{\mathrm{UL}} + E_i^{\mathrm{LC}} \leq E_i^{\mathrm{EH}} \; \forall i \in \mathcal{I}$$

In particular, the number of sub-channels for computating offloading UE must be greater than 2, while local computation of UE at least require 1 sub-channel, which can be expressed as the following constraint

$$C11: (1 - \sum_{k \in \mathcal{K}} b_{i,k}) \ln \left\{\sum_{n \in \mathcal{N}} w_{i,n}\right\} = 0 \quad \forall i \in \mathcal{I}$$

$$C12: 0 \leq \sum_{k \in \mathcal{K}} b_{i,k} \ln \left\{\frac{\sum_{n \in \mathcal{N}} w_{i,n}}{2}\right\} \quad \forall i \in \mathcal{I}$$

To maximize the energy efficiency of the SWIPT-based MEC network and provide enough energy for the UE's computing offload or local computing, we jointly optimize the sub-channel allocation, offloading strategy, UE's calculation frequency and uplink power in this paper. For simplicity, we assume that the energy consumption of computing tasks is much higher than that of optimized calculations, and the energy consumption of sending and receiving devices is ignored [32]. Based on the above analysis and reference [33], [34], the objective function can be expressed as $\mathbf{P1}$. Since the network is binary offloading, the total amount of data flowing $\sum_{k \in \mathcal{K}} b_{i,k} D_i$ in the network is equal to the tasks offloaded by the UE.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3047690, IEEE Access

$$
\begin{aligned}
\mathbf{P1}: \max_{P_{i,n}^{\mathrm{UL}}, w_{i,n}, b_{i,k}, F_i^{\mathrm{LC}}} & \sum_{i\in\mathcal{I}} \frac{\sum_{k\in\mathcal{K}} b_{i,k} D_i}{E_i^{\mathrm{UL}} + E_i^{\mathrm{LC}}} \\
\text{s.t. } & C1: b_{i,k} \in \{0,1\} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K} \\
& C2: w_{i,n} \in \{0,1\} \quad \forall i \in \mathcal{I}, \forall n \in \mathcal{N} \\
& C3: \sum_{i\in\mathcal{I}} w_{i,n} \le 1 \quad \forall n \in \mathcal{N} \\
& C4: \sum_{k\in\mathcal{K}} b_{i,k} \le 1 \quad \forall i \in \mathcal{I} \\
& C5: 0 \le \sum_{i\in\mathcal{I}} F_{i,k}^{\mathrm{UC}} \le F_k \quad \forall k \in \mathcal{K} \\
& C6: 0 \le F_i^{\mathrm{LC}} \le F_i \quad \forall i \in \mathcal{I} \\
& C7: \sum_{n\in\mathcal{N}} P_{i,n}^{\mathrm{UL}} \le P_i^{\mathrm{Max}} \quad \forall i \in \mathcal{I} \\
& C8: T_i^{\mathrm{LC}} \le t_i^{\mathrm{tolerant}} \quad \forall i \in \mathcal{I} \\
& C9: T_i^{\mathrm{UC}} + T_i^{\mathrm{UL}} \le t_i^{\mathrm{tolerant}} \quad \forall i \in \mathcal{I} \\
& C10: E_i^{\mathrm{UL}} + E_i^{\mathrm{LC}} \le E_i^{\mathrm{EH}} \; \forall i \in \mathcal{I} \\
& C11: (1 - \sum_{k\in\mathcal{K}} b_{i,k}) \ln \left\{ \sum_{n\in\mathcal{N}} w_{i,n} \right\} = 0 \quad \forall i \in \mathcal{I} \\
& C12: 0 \le \sum_{k\in\mathcal{K}} b_{i,k} \ln \left\{ \frac{\sum_{n\in\mathcal{N}} w_{i,n}}{2} \right\} \quad \forall i \in \mathcal{I}
\end{aligned} \tag{9}
$$

III. SIMPLIFICATION OF THE ENERGY EFFICIENCY OPTIMIZATION MODEL

The energy efficiency optimization model proposed in the previous section is MINLP and non-convex. This section will mathematically simplify the original model to make the problem solvable and decrease complexity.

For simplify the objective function, we divide the numerator and denominator by T^{UL}. After the operation, $\mathbf{P1}$ becomes $\mathbf{P2}$.

$$
\begin{aligned}
\mathbf{P2}: & \\
& \max_{P_{i,n}^{\mathrm{UL}}, w_{i,n}, b_{i,k}, F_i^{\mathrm{LC}}} \sum_{i\in\mathcal{I}} \frac{\sum_{k\in\mathcal{K}} b_{i,k} \sum_{n\in\mathcal{N}} (w_{i,n} - \frac{1}{N}) R_{i,n}^{\mathrm{UL}}}{\sum_{n\in\mathcal{N}} P_{i,n}^{\mathrm{UL}} + \frac{E_i^{\mathrm{LC}}}{T_i^{\mathrm{UL}}}} \\
& \text{s.t. } C1 \sim 12
\end{aligned} \tag{10}
$$

where

$$
\frac{E_i^{\mathrm{LC}}}{T_i^{\mathrm{UL}}} = (1 - \sum_{k\in\mathcal{K}} b_{i,k}) k_0 \left(F_i^{\mathrm{LC}}\right)^2 X_i \sum_{n\in\mathcal{N}} (w_{i,n} - \frac{1}{N}) R_{i,n}^{\mathrm{UL}} \tag{11}
$$

In the case of local computing, the term $\left(1 - \sum_{k\in\mathcal{K}} b_{i,k}\right)$ is always equal to zero. Else, $\sum_{n\in\mathcal{N}} (w_{i,n} - \frac{1}{N})$ is always 0. Therefore, (11) has no effect on the denominator and can be deleted. In the same way, the $\sum_{k\in\mathcal{K}} b_{i,k}$ in (10) can also be removed. At this time, the practical meaning of the objective function is to maximize the energy efficiency of the UE of computation offloading , regardless of the UE of local computing. This will cause an imbalance between UEs. To provides the best services to all UEs, we added energy efficiency of the UE of local computing to the objective function, as shown in $\mathbf{P3}$.

$$
\begin{aligned}
\mathbf{P3}: & \max_{P_{i,n}^{\mathrm{UL}}, w_{i,n}, b_{i,k}, F_i^{\mathrm{LC}}} \sum_{i\in\mathcal{I}} \left(\frac{\sum_{n\in\mathcal{N}} (w_{i,n} - \frac{1}{N}) R_{i,n}^{\mathrm{UL}}}{\sum_{n\in\mathcal{N}} P_{i,n}^{\mathrm{UL}}} + \frac{(1 - \sum_{k\in\mathcal{K}} b_{i,k}) D_i}{E_i^{\mathrm{LC}}} \right) \\
& \text{s.t. } C1 \sim 12
\end{aligned} \tag{12}
$$

In $\mathbf{P3}$, the value of T_i^{UL} is influenced by $t_i^{\mathrm{tolerant}} - T_i^{\mathrm{UC}}$, and T_i^{UC} is constrained by $t_i^{\mathrm{tolerant}} - T_i^{\mathrm{UL}}$. Obviously, $C12$ is a constraint that has physical meaning but is not solvable. So we use new constraints replace $C12$ to limit the minimum value of R^{UL}. The modified model is as follows:

$$
\begin{aligned}
\mathbf{P3'}: & \max_{P_{i,n}^{\mathrm{UL}}, w_{i,n}, b_{i,k}, F_i^{\mathrm{LC}}} \sum_{i\in\mathcal{I}} \left(\frac{\sum_{n\in\mathcal{N}} (w_{i,n} - \frac{1}{N}) R_{i,n}^{\mathrm{UL}}}{\sum_{n\in\mathcal{N}} P_{i,n}^{\mathrm{UL}}} + \frac{(1 - \sum_{k\in\mathcal{K}} b_{i,k}) D_i}{E_i^{\mathrm{LC}}} \right) \\
& \text{s.t. } C1 \sim 8, 10 \sim 12 \\
& \quad C9: R_{i,n}^{\min} \le R_{i,n}^{\mathrm{UL}} \quad \forall i \in \mathcal{I}, \forall n \in \mathcal{N}
\end{aligned} \tag{13}
$$

where $R_{i,n}^{\min}$ is the minimum transmission rate of each sub-channel. The reason why this paper does not choose to limit the overall rate of UE is to make full use of each sub-channel. As the number of sub-channels held by UE increases, the transmission rate of each sub-channel will be decreased while the overall speed remains unchanged. Using subchannels in this way is very luxurious.

IV. THREE-PART ALTERNATE OPTIMIZATION ALGORITHM FRAMEWORK

In this section, the optimization algorithm framework proposed in this paper will be introduced in detail. In order to accurately match variables and algorithms, this paper classifies the optimized variables. Since continuous variables and discrete variables need to be processed by different algorithms, P^{UL} and F^{LC} are regarded as one part, and B and W are classified as another. However, the optimization of W needs to depend on P^{UL} and B. Therefore B and W have to be handled separately. Since F^{LC} belongs to hardware information and is only related to B, the UE can obtain the optimal F^{LC} from the MEC and store it after uploading the hardware information. When $b_{i,k}$ = 0, the CPU works with the optimal F^{LC}, else, not work. Whatmore, the other three variables are dependent on each other. So this paper uses a three-part alternating optimization method to optimize them. The algorithm framework is shown in Fig. 2:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3047690, IEEE Access

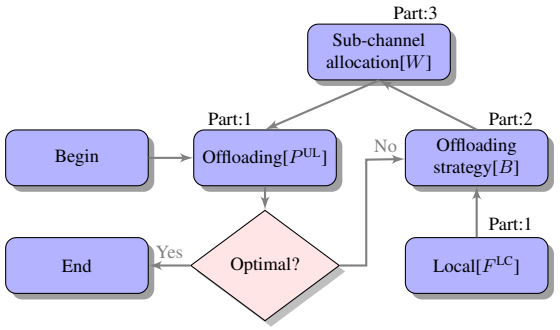

FIGURE 2. The three-part alternate optimization algorithm framework.

First, the submodels of part one can be written as $\mathbf{P3'-a}$ and $\mathbf{P3'-b}$.

$$\mathbf{P3'-a}: \max_{F_i^{\mathrm{LC}}} \sum_{i\in\mathcal{I}} \left[\frac{1-\sum_{k\in\mathcal{K}} \bar{b}_{i,k}}{k_0 X (F_i^{\mathrm{LC}})^2} \right] \quad \text{s.t. } C6, C8, C10 \tag{14}$$

$$\mathbf{P3'-b}: \max_{P_{i,n}^{\mathrm{UL}}} \sum_{i\in\mathcal{I}} \left[\frac{\sum_{n\in\mathcal{N}} (\bar{w}_{i,n} - \frac{1}{N}) R_{i,n}^{\mathrm{UL}}}{\sum_{n\in\mathcal{N}} P_{i,n}^{\mathrm{UL}}} \right] \quad \text{s.t. } C7, C9 \sim 10 \tag{15}$$

Here, $\bar{b}_{i,k}$ and $\bar{w}_{i,n}$ are the current optimal solutions obtained from the other two parts. The first part uses Dinkelbach algorithm to transform the objective function of each sub-model into a polynomial. Then we combine the ADMM algorithm and KKT conditions to optimize them. After completing the first part, the framework will determine whether the change in energy efficiency is less than the preset accuracy. If yes, return the result, otherwise, enter the second part. And Algorithm 1 is used in part two to complete the offloading decision. Finally, the third part uses integer bat algorithm to get the sub-channel allocation strategy and returns the result to the first part. Repeat like that until the optimal solution is obtained.

Algorithm 1 Algorithm of computation offloading strategy

Initialize $i = 0$;
for $i \in \mathcal{I}$ **do**
  if $E_i^{\mathrm{UL}} < E_i^{\mathrm{LC}}$ **then**
    UE_i offloads tasks to corresponding MEC_k;
    $F_k^{\mathrm{UC}} = F_k^{\mathrm{UC}} + \frac{D_i X}{t_i^{\mathrm{tolerant}} - T_i^{\mathrm{UL}}}$;
    if $F_k^{\mathrm{UC}} \le F_k$ **then**
      Force UE_i local computing;
    end if
  else
    Local computing;
  end if
end for

A. PART-1: DINKELBACH

Dinkelbach is an algorithm for fractional objective programming. Its main principle is to approach the optimal value step by step through iteration. Obviously, the objective function of this paper is also fractional, so we use this algorithm to get a solution. According to the description in [35], $\mathbf{P3'-a}$ and $\mathbf{P3'-b}$ can be converted into

$$\mathbf{P4a}: \max_{F_i^{\mathrm{LC}}} \sum_{i\in\mathcal{I}} \left[(1 - \sum_{k\in\mathcal{K}} \bar{b}_{i,k}) - q_i^{\mathrm{F}} k_0 X (F_i^{\mathrm{LC}})^2 \right] \quad \text{s.t. } C6, C8, C10 \tag{16}$$

$$\mathbf{P4b}: \max_{P_{i,n}^{\mathrm{UL}}} \sum_{i\in\mathcal{I}} \left[\sum_{n\in\mathcal{N}} (\bar{w}_{i,n} - \frac{1}{N}) R_{i,n}^{\mathrm{UL}} - q_i^{\mathrm{P}} \sum_{n\in\mathcal{N}} P_{i,n}^{\mathrm{UL}} \right] \quad \text{s.t. } C7, C9 \sim 10 \tag{17}$$

q_i^{P} and q_i^{F} is the auxiliary variable led into by the algorithm, which will gradually increase with the advance of iteration and tend to be flat. We define the error judgment function of $\mathbf{P4a}$ and $\mathbf{P4b}$ as $F(F_i^{\mathrm{LC}})$ and $F(P_{i,n}^{\mathrm{UL}})$

$$F(F_i^{\mathrm{LC}}) = \sum_{i\in\mathcal{I}} \left(q_i^F[j] - q_i^F[j-1] \right) \tag{18}$$

$$F(P_{i,n}^{\mathrm{UL}}) = \sum_{i\in\mathcal{I}} \left(q_i^P[j] - q_i^P[j-1] \right) \tag{19}$$

which is used to evaluate the accuracy of the calculation result. Their values will slowly decrease until the accuracy requirements are met. The specific process of using Dinkelbach algorithm is presented in Algorithm 2. Where, J is the maximum of iterations and ϵ_D is the convergence tolerance.

Algorithm 2 Dinkelbach algorithm

Initialize $j = 0$;
• Local Computing
while $j \le J$ and $F(F_i^{\mathrm{LC}}) \le \epsilon_D$ **do**
  Solve for $\mathbf{P4a}$ to get the optimal solution $\{F^{\mathrm{LC}}\}$;
  $q_i^{\mathrm{F}} = \sum_{i\in\mathcal{I}} \left[\frac{1-\sum_{k\in\mathcal{K}} \bar{b}_{i,k}}{k_0 X (F_i^{\mathrm{LC}})^2} \right]$;
  $j = j + 1$;
end while
• Computation Offloading
while $j \le J$ and $F(P_{i,n}^{\mathrm{UL}}) \le \epsilon_D$ **do**
  Solve for $\mathbf{P4b}$ to get the optimal solution $\{P^{\mathrm{UL}}\}$;
  $q_i^{\mathrm{P}} = \left[\frac{\sum_{n\in\mathcal{N}} (\bar{w}_{i,n} - \frac{1}{N}) R_{i,n}^{\mathrm{UL}}}{\sum_{n\in\mathcal{N}} P_{i,n}^{\mathrm{UL}}} \right]$;
  $j = j + 1$;
end while

B. PART-1: ADMM

ADMM is widely used in solving constrained optimization problems. It is easy to decompose and converges well on medium and low precision problems. The ADMM algorithm divides the original problem into two or more branches by introducing auxiliary variables to perform alternate optimization. Refer to [36]–[38], this paper created a new variable $\check{P^{\mathrm{UL}}}$ as a copy of P^{UL}, similarly, $\check{F^{\mathrm{LC}}}$ corresponds to F^{LC}, which makes it possible to alternately optimize $\mathbf{P4a}$ and $\mathbf{P4b}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3047690, IEEE Access

$$L_F^1(F_i^{\mathrm{LC}}) = \sum_{i\in\mathcal{I}}\Big(\sum_{k\in\mathcal{K}}\bar{b}_{i,k} - 1 + q_i^{\mathrm{F}}k_0X(F_i^{\mathrm{LC}})^2 + \lambda_i^{\mathrm{F}}(F_i^{\mathrm{LC}} - \check{F}_i^{\mathrm{LC}}) + \frac{\rho_F}{2}\|F_i^{\mathrm{LC}} - \check{F}_i^{\mathrm{LC}}\| + \gamma_i^{\mathrm{F2}}(E_i^{\mathrm{LC}} - E_i^{\mathrm{EH}}) + \gamma_i^{\mathrm{F3}}(D_iX - F_i^{\mathrm{LC}}t_i^{\mathrm{tolerant}})\Big) \tag{23}$$

$$L_F^2(\check{F}_i^{\mathrm{LC}}) = \sum_{i\in\mathcal{I}}\Big(\lambda_i^{\mathrm{F}}(F_i^{\mathrm{LC}} - \check{F}_i^{\mathrm{LC}}) + \frac{\rho_F}{2}\|F_i^{\mathrm{LC}} - \check{F}_i^{\mathrm{LC}}\| + \gamma_i^{\mathrm{F1}}(F_i^{\mathrm{LC}} - F_i^{\max})\Big) \tag{24}$$

1) Local Computing

The **P4a** can be written as

$$\begin{aligned}\mathbf{P5a.1}: \min_{F_i^{\mathrm{LC}}} & \sum_{i\in\mathcal{I}}\Big(\sum_{k\in\mathcal{K}}\bar{b}_{i,k} - 1 + q_i^{\mathrm{F}}k_0X(F_i^{\mathrm{LC}})^2 + \\ & \lambda_i^{\mathrm{F}}(F_i^{\mathrm{LC}} - \check{F}_i^{\mathrm{LC}}) + \frac{\rho_F}{2}\|F_i^{\mathrm{LC}} - \check{F}_i^{\mathrm{LC}}\|\Big) \\ \text{s.t. } & C8: D_iX \le F_i^{\mathrm{LC}}t_i^{\mathrm{tolerant}}\ \forall i\in\mathcal{I} \\ & C10: E_i^{\mathrm{LC}} \le E_i^{\mathrm{EH}}\ \forall i\in\mathcal{I}\end{aligned} \tag{20}$$

$$\begin{aligned}\mathbf{P5a.2}: \min_{\check{F}_i^{\mathrm{LC}}} & \sum_{i\in\mathcal{I}}\Big(\lambda_i^{\mathrm{F}}(F_i^{\mathrm{LC}} - \check{F}_i^{\mathrm{LC}}) + \frac{\rho_F}{2}\|F_i^{\mathrm{LC}} - \check{F}_i^{\mathrm{LC}}\|\Big) \\ \text{s.t. } & C6: 0 \le \check{F}_i^{\mathrm{LC}} \le F_i^{\max} \quad \forall i\in\mathcal{I}\end{aligned} \tag{21}$$

here, we give $C6$ to the auxiliary variable $\check{F}^{\mathrm{LC}}$, and leave the rest to F^{LC}. $C8$ moved F_i^{LC} from the left to the right, so as to avoid fractional calculations and facilitate subsequent processing. The two optimization variables cooperate to optimize the problem alternately. In the process of optimization, the two variables will become closer and closer. This is because the penalty function in the objective function is at work. ρ_F is the penalty coefficient, which represents the sensitivity to the difference between the two variables. In addition, λ_i^{F} is the Lagrangian coefficient, which is obtained by the constraints $F_i^{\mathrm{LC}} = \check{F}_i^{\mathrm{LC}}$ introduced by the ADMM algorithm. Its update formula is as follows.

$$\lambda_i^{\mathrm{F}}[t+1] = \lambda_i^{\mathrm{F}}[t] + \rho_F(F_i^{\mathrm{LC}} - \check{F}_i^{\mathrm{LC}}) \tag{22}$$

For λ_i^{F}, the value of $t+1$ iterations is determined by the value of t iterations and the difference between the two variables. For F_i^{LC} and $\check{F}^{\mathrm{LC}}$, KKT is used to solve it in this paper. According to the three constraints of **P5a.1** and **P5a.2**, the Lagrangian function $L_F^1(F_i^{\mathrm{LC}})$ and $L_F^2(\check{F}_i^{\mathrm{LC}})$ can be written as (23) and (24).

The KKT condition introduces that the partial derivative of the Lagrangian function should tend to 0 when reaching the optimum. Thus, update method of F_i^{LC} and $\check{F}_i^{\mathrm{LC}}$ can be obtained by calculating the partial derivative of the two variables respectively. The specific expression is:

$$F_i^{\mathrm{LC}} = \left[\frac{-\lambda_i^{\mathrm{F}} + t_i^{\mathrm{tolerant}}\gamma_i^{\mathrm{F3}} + \rho_F\check{F}_i^{\mathrm{LC}}}{\rho_F + 2Xk_0(D_i\gamma_i^{\mathrm{F2}} + q_i^{\mathrm{F}})}\right]^+ \tag{25}$$

$$\check{F}_i^{\mathrm{LC}} = \left[\frac{\lambda_i^{\mathrm{F}} - \gamma_i^{\mathrm{F1}}}{\rho_F} + F_i^{\mathrm{LC}}\right]^+ \tag{26}$$

where $[\bullet]^+$ limits the range of $\bullet$, which can only be $[0, +\infty)$. $\gamma_i^{\mathrm{F1}}, \gamma_i^{\mathrm{F2}} and \gamma_i^{\mathrm{F3}}$ represent the Lagrangian coefficient of constraint $C6, C8, C10$. Similarly, the update algorithm of the introduced auxiliary variable is as follows

$$\gamma_i^{\mathrm{F1}} = \left[\gamma_i^{\mathrm{F1}} + \nabla_{F1}(\check{F}_i^{\mathrm{LC}} - F_i^{\max})\right]^+ \tag{27}$$

$$\gamma_i^{\mathrm{F2}} = \left[\gamma_i^{\mathrm{F2}} + \nabla_{F2}(E_i^{\mathrm{LC}} - E_i^{\mathrm{EH}})\right]^+ \tag{28}$$

$$\gamma_i^{\mathrm{F3}} = \left[\gamma_i^{\mathrm{F3}} + \nabla_{F3}(D_iX - F_i^{\mathrm{LC}}t_i^{\mathrm{tolerant}})\right]^+ \tag{29}$$

where, $\nabla_{F1}, \nabla_{F2}, \nabla_{F3}$ is the weight of the update step.

2) Computation Offloading

Same as above, model **P4b** can become **P5b**.

$$\begin{aligned}\mathbf{P5b.1}: \min_{P_{i,n}^{\mathrm{UL}}} & \sum_{i\in\mathcal{I}}\Big(-\sum_{n\in\mathcal{N}}(\bar{w}_{i,n} - \frac{1}{N})R_{i,n}^{\mathrm{UL}} + q_i^{\mathrm{P}}\sum_{n\in\mathcal{N}}P_{i,n}^{\mathrm{UL}} \\ & + \lambda_i^{\mathrm{P}}\sum_{n\in\mathcal{N}}(P_{i,n}^{\mathrm{UL}} - \check{P}_{i,n}^{\mathrm{UL}}) + \frac{\rho_P}{2}\sum_{n\in\mathcal{N}}\|P_{i,n}^{\mathrm{UL}} - \check{P}_{i,n}^{\mathrm{UL}}\|\Big) \\ \text{s.t. } & C9: R_{i,n}^{\min} \le R_{i,n}^{\mathrm{UL}} \quad \forall i\in\mathcal{I}, \forall n\in\mathcal{N} \\ & C10: E_i^{\mathrm{UL}} \le E_i^{\mathrm{EH}}\ \forall i\in\mathcal{I}\end{aligned} \tag{30}$$

$$\begin{aligned}\mathbf{P5b.2}: \min_{\check{P}_{i,n}^{\mathrm{UL}}} & \sum_{i\in\mathcal{I}}\Big(\lambda_i^{\mathrm{P}}\sum_{n\in\mathcal{N}}(P_{i,n}^{\mathrm{UL}} - \check{P}_{i,n}^{\mathrm{UL}}) + \\ & \frac{\rho_P}{2}\sum_{n\in\mathcal{N}}\|P_{i,n}^{\mathrm{UL}} - \check{P}_{i,n}^{\mathrm{UL}}\|\Big) \\ \text{s.t. } & C7: \check{P}_{i,n}^{\mathrm{UL}} \le P_{i,n}^{\mathrm{Max}} \quad \forall i\in\mathcal{I}\end{aligned} \tag{31}$$

Using the same algorithm, let $\check{P}_{i,n}^{\mathrm{UL}}$ satisfy $C7$ and allocate the rest to $P_{i,n}^{\mathrm{UL}}$. ρ_P is the coefficient of the penalty function in the computation offloading model. λ_i^{P} is the Lagrangian coefficient corresponding to the constraint $P_{i,n}^{\mathrm{UL}} = \check{P}_{i,n}^{\mathrm{UL}}$, and the update formula is

$$\lambda_i^{\mathrm{P}}[t+1] = \lambda_i^{\mathrm{P}}[t] + \rho_P\sum_{n\in\mathcal{N}}(P_{i,n}^{\mathrm{UL}} - \check{P}_{i,n}^{\mathrm{UL}}) \tag{32}$$

In this part, KKT conditions are still used to solve each branch of alternate optimization. Formula (33) and (34) show the Lagrangian functions $L_P^1(P_{i,n}^{\mathrm{UL}})$ and $L_P^2(\check{P}_{i,n}^{\mathrm{UL}})$ corresponding to **P5b.1** and **P5b.2**.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3047690, IEEE Access

$$L_P^1(P_{i,n}^{\mathrm{UL}}) = \sum_{i\in\mathcal{I}}\Big(-\sum_{n\in\mathcal{N}}(w_{i,n}-\frac{1}{N})R_{i,n}^{\mathrm{UL}} + q_i^{\mathrm{P}}\sum_{n\in\mathcal{N}}P_{i,n}^{\mathrm{UL}} + \lambda_i\sum_{n\in\mathcal{N}}(P_{i,n}^{\mathrm{UL}}-\check{P}_{i,n}^{\mathrm{UL}}) + \frac{\rho_p}{2}\sum_{n\in\mathcal{N}}\|P_{i,n}^{\mathrm{UL}}-\check{P}_{i,n}^{\mathrm{UL}}\| + \sum_{n\in\mathcal{N}}\gamma_{i,n}^{\mathrm{P2}}(R_{i,n}^{\min}-R_{i,n}^{\mathrm{UL}}) + \gamma_i^{\mathrm{P3}}(\sum_{n\in\mathcal{N}}P_{i,n}^{\mathrm{UL}} - \frac{\sum_{n\in\mathcal{N}}(w_{i,n}-\frac{1}{N})R_{i,n}^{\mathrm{UL}}E_i^{\mathrm{EH}}}{D_i})\Big) \tag{33}$$

$$L_P^2(\check{P}_{i,n}^{\mathrm{UL}}) = \sum_{i\in\mathcal{I}}\Big(\lambda_i\sum_{n\in\mathcal{N}}(P_{i,n}^{\mathrm{UL}}-\check{P}_{i,n}^{\mathrm{UL}}) + \frac{\rho_p}{2}\sum_{n\in\mathcal{N}}\|P_{i,n}^{\mathrm{UL}}-\check{P}_{i,n}^{\mathrm{UL}}\| + \sum_{n\in\mathcal{N}}\gamma_{i,n}^{\mathrm{P1}}(\check{P}_{i,n}^{\mathrm{UL}}-P_{i,n}^{\mathrm{Max}})\Big) \tag{34}$$

By calculating the partial derivative, the update formula of $P_{i,n}^{UL}$ and $\check{P}_{i,n}^{UL}$ can be denoted as

$$P_{i,n}^{\mathrm{UL}} = \left[\sqrt{\frac{X^2}{4}-\frac{G}{\rho_P\sum_{n\in\mathcal{K}}\bar{b}_{i,k}h_{i,n,k}}}-\frac{X}{2}\right]^+ \tag{35}$$

$$\check{P}_{i,n}^{\mathrm{UL}} = \left[P_{i,n}^{\mathrm{UL}}+\frac{\lambda_i-\gamma_{i,n}^{\mathrm{P1}}}{\rho_P}\right]^+ \tag{36}$$

where

$$X = \frac{\lambda_i+q_i^{\mathrm{P}}+\gamma_i^{\mathrm{P3}}}{\rho_P} + \frac{\sigma^2}{\sum_{n\in\mathcal{K}}\bar{b}_{i,k}h_{i,n,k}N} - \check{P}_{i,n}^{\mathrm{UL}} \tag{37}$$

$$G = (1+\gamma_{i,n}^{\mathrm{P2}}+\frac{\gamma_i^{\mathrm{P3}}E_i^{\mathrm{EH}}}{D_i})\sum_{n\in\mathcal{K}}\bar{b}_{i,k}h_{i,n,k}(\bar{w}_{i,n}-\frac{1}{N})\frac{B_N}{\ln 2} + \frac{\sigma^2(\lambda_i+q_i^{\mathrm{P}}-\check{P}_{i,n}^{\mathrm{UL}}\rho_p+\gamma_i^{\mathrm{P3}})}{N} \tag{38}$$

$\gamma_{i,n}^{\mathrm{P1}}$, $\gamma_{i,n}^{\mathrm{P2}}$ and γ_i^{P3} represent the Lagrangian coefficient of constraint $C7, C9\sim 10$, and iterative formula can be written as

$$\gamma_{i,n}^{\mathrm{P1}} = \left[\gamma_{i,n}^{\mathrm{P1}}+\nabla_{P1}(\check{P}_{i,n}^{\mathrm{UL}}-P_{i,n}^{\mathrm{Max}})\right]^+ \tag{39}$$

$$\gamma_{i,n}^{\mathrm{P2}} = \left[\gamma_{i,n}^{\mathrm{P2}}+\nabla_{P2}(R_{i,n}^{\min}-R_{i,n}^{\mathrm{UL}})\right]^+ \tag{40}$$

$$\gamma_i^{\mathrm{P3}} = \left[\gamma_i^{\mathrm{P3}}+\nabla_{P3}(\sum_{n\in\mathcal{N}}P_{i,n}^{\mathrm{UL}}-\frac{\sum_{n\in\mathcal{N}}(\bar{w}_{i,n}-\frac{1}{N})R_{i,n}^{\mathrm{UL}}E_i^{\mathrm{EH}}}{D_i})\right]^+ \tag{41}$$

also, $\nabla_{P1}, \nabla_{P2}, \nabla_{P3}$ is the weight of the update step.

C. PART-3: INTEGER BAT ALGORITHM

For W, which is an integer variable of 0 or 1, the common method is to turn it into a continuous variable to facilitate the solution. But the authenticity is weakened by approximation. In [39], although the final result is only 0 or 1, the process of allocating sub-channels needs to calculate the gradient with the help of KKT conditions. This is still distorted. Bat algorithm is a meta-heuristic optimization algorithm [40] and is superior to the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithm. The Bat algorithm mainly imitates the hunting of bats and achieves the optimal solution by continuously moving to the temporary optimal bat. It does not need to calculate the derivative, so we choose to develop a new algorithm based on the bat algorithm.

Algorithm 3 Solve ADMM model with KKT condition

Input: J_{ADMM}: maximum of iterations;
Output: Optimal F^{LC} and P^{UL}

1: Initialize $j=0$;
• Local Computing
2: **while** $j \le J_{ADMM}$ **do**
3: Branch 1: Update F_i^{LC} and $\gamma_i^{\mathrm{F2}}, \gamma_i^{\mathrm{F3}}$ with (25,28,29);
4: Branch 2: Update $F_i^{\mathrm{LC}}, \gamma_i^{\mathrm{F1}}$ with (26,27);
5: Update λ_i^{F} with (22);
6: $j = j+1$;
7: **end while**
• Computation Offloading
8: **while** $j \le J_{ADMM}$ **do**
9: Branch 1: Update $P_{i,n}^{\mathrm{UL}}$ and $\gamma_{i,n}^{\mathrm{P2}}, \gamma_i^{\mathrm{P3}}$ with (35,40,41);
10: Branch 2: Update $\check{P}_{i,n}^{\mathrm{UL}}, \gamma_{i,n}^{\mathrm{P1}}$ with (36,39);
11: Update λ_i^{P} with (32);
12: $j = j+1$;
13: **end while**

First, the prey of the bats needs to be expressed, which is the objective function. This function is composed of all polynomials related to W in $\mathbf{P5b.1}$ and $\mathbf{P5b.2}$, to jointly optimize $P_{i,n}^{\mathrm{UL}}$ and $\check{P}_{i,n}^{\mathrm{UL}}$, it can be expressed as

$$L_W = \sum_{i\in\mathcal{I}}\left[-(1+\gamma_{i,n}^{\mathrm{P2}}+\frac{\gamma_i^{\mathrm{P3}}E_i^{\mathrm{EH}}}{D_i})\sum_{n\in\mathcal{N}}(w_{i,n}-\frac{1}{N})R_{i,n}^{\mathrm{UL}}\right] \tag{42}$$

We think of W as a bat. Since it is $\sum_{n\in\mathcal{N}}(w_{i,n}-\frac{1}{N})$ instead of W that has effect on the value of L_W, the number of sub-channels allocated to each UE can be regarded as the positions of the bats. The process of the suboptimal bat approaching the optimal bat becomes a process of a 20-dimensional vector approaching another 20-dimensional vector. Based on this principle, we divide UEs into two categories according to the number of sub-channels held. The classification standard can be expressed as

$$\begin{cases} UE_i \in N_{big} & N[UE_i]^W > N[UE_i]^{W_{best}} \\ UE_i \in N_{small} & N[UE_i]^W < N[UE_i]^{W_{best}} \end{cases} \tag{43}$$

where $N[\bullet]^W$ indicates the number of sub-channels obtained by UE_i under the sub-channel allocation strategy W. W_{best} represents the temporary optimal strategy. Since the position

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3047690, IEEE Access

of the bat is 20-dimensional, the speed at which each bat approaches the optimal bat can be defined as a 20-dimensional array $V = \{V_i \mid \forall i \in \mathcal{I}\}$. The algorithm is as follows

$$V_i = \left[Q_i(N[UE_i]^W - N[UE_i]^{W_{best}})\right]_{ceil} \tag{44}$$

here, $Q = \{Q_i \mid \forall i \in \mathcal{I}\}$ is a positive random number less than 1, meaning the vibration frequency of the bat. And the function $[\bullet]_{ceil}$ rounds the value $\bullet$ to the smallest integer not less than it. After all bats have moved one step, it will be judged whether the current optimal strategy is still optimal. For convenience, we assume that the UE_i choosing computation offloading is classified as UE^{O}.

We kept the dithering operation of the bat algorithm during the optimization process. In order to prevent the optimization result from being a sub-optimal solution, it is possible to randomly change the optimal solution during each iteration. And only when the modified optimal value is greater than the original, the optimal strategy will be updated. The probability of jitter increases as the iteration progresses, and the movement speed and frequency of the bat will gradually decrease until the optimal value is obtained. The entire algorithm has been shown in Algorithm 4.

D. COMPUTATIONAL COMPLEXITY

This subsection will provide the computational complexity of the algorithm. By analyzing Algorithm 2 and Algorithm 3, the computational complexity of the first part is $\mathcal{O}(INK)$. Similarly, the complexity of the second part is $\mathcal{O}(I)$, and the third part is $\mathcal{O}(INK)$. Suppose that the three-part alternate optimization requires ψ cycles. The computational complexity of this algorithm framework is $\mathcal{O}(\psi INK)$. However in our network, the computational complexity of the algorithm in [39] is $\mathcal{O}(I^2NK)$.

V. SIMULATION AND RESULT ANALYSIS

TABLE 1. Simulation Parameters

Parameters	Value
Subchannel bandwidth B_N	12.5 KHz
Noise σ^2	$10^{(-13)}$ W
Bit length of the task D_i	$1000 \sim 1500$ bit
Operational capability X	1000 cycles/bit
Maximum transmission power $P_{i,n}^{\mathrm{Max}}$	$0.5 \sim 0.6$ W
Downlink power $P_{i,k}^{\mathrm{DL}}$	$2.5 \sim 2.6$ W
Tolerance time t_i^{tolerant}	$9 \sim 10$ ms
The maximum CPU frequency of UE F_i	0.2GHz
The maximum CPU frequency of MEC F_k	0.5GHz
Random Raylaigh Fading Channel gain $h_{i,n}$	$\mathcal{CN}(0, 0.01)$

In this part, we give the experimental results and make further analysis, the parameter settings refer to [41], [42] and are listed in the Tab. 1. In addition, we also specified the number of devices and sub-channels in the SWIPT-based MEC network, let $\mathcal{I} = 20, \mathcal{K} = 7, \mathcal{N} = 128$. The initialization before the algorithm starts is as follows: when calculating the local computing area and computation offloading area, except that W and B initially require a random strategy, the other initial values are all zero. Algorithm 1 does not require initial settings. For Algorithm 4, let $A = r = \mathcal{N}$ and $sizep = 50$.

Algorithm 4 0-1 integer bat algorithm

Input: $sizep$: Number of bats;
A/r: Sonic loudness/pulse of bat (size: sizep $\times$ 1);
A_f/R_f: Loudness/pulse update factor;
ϵ: Maximum error;
J_{max}:The maximum number of iterations;
τ:Dithering coefficient;
Output: Optimal W;

Reference B randomly allocates sub-channels to all bats;
According to (42) find the best bat W_{best};
while $[j < J_{max}]$ and $[L_W^{[t-1]} - L_W^{[t]} \geq \epsilon]$ **do**
  for each $s \in [1, sizep]$ **do**
    for each $UE_m \in \{N_{big}\}$ **do**
      UE_m give V sub-channels to $UE_g \in \{N_{small}\}$ $(m \neq g)$;
    end for
    $W_{Stemp} = W_s$; $i_s = 0$;
    Generate a random number $m \in [1, \mathcal{N}]$;
    if $m \geq r$ **then**
      $W_{Stemp} = W_{best}$;
      repeat
        For $UE \in UE^{\mathrm{O}}$, randomly select UE_a to give UE_b a sub-channel $(a \neq b)$
        $i_s + 1$;
        if $N[UE_a]^{W_{Stemp}} < 2$ **then**
          Return this UE_a's sub-channel
        end if
      until $i_s > \tau size(UE^{\mathrm{O}})$;
    end if
  end for
  Calculate L_W^{S} with W_{Stemp} by (42) ;
  if $m < A$ and $L_W^{\mathrm{S}} < L_W$ **then**
    Let $W_s = W_{Stemp}$ and $L_W = L_W^{\mathrm{S}}$
    $A = A_f A$;
    $r = r_0 * (1 - e^{-R_f j})$
  end if
  if $L_W^{\mathrm{S}} < L_{min}$ **then**
    Let $W_{best} = W_{Stemp}$ and $L_{min} = L_W^{\mathrm{S}}$
  end if
  j = j+1;
end while
return $W = W_{best}$

Fig. 3 shows the impact of varying number of UEs on energy efficiency of SWIPT-based MEC Networks. It can be found from the Fig. 3 that the energy efficiency of the network increases with the increase of the number of UEs. This is because the energy efficiency is sum of that of each UE. It proves that our algorithm is capable of serving more UEs and is suitable for green communication.

We evaluate the energy efficiency of our proposed algorithm and compare it with the algorithm of [39], computation offloading only and local computing only. Given the fixed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3047690, IEEE Access

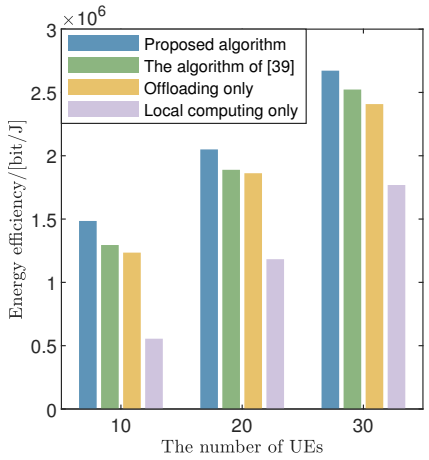

FIGURE 3. Comparison of energy efficiency with varying number of UEs.

value of $\mathcal{I}$, the result of our proposed algorithm is the best one. When $\mathcal{I} = 20$, the energy efficiency of our proposed algorithm is 11% higher than the algorithm of [39]. When not considering computation offloading, the energy efficiency is lower than the one of considering computation offloading, so binary offloading can improve the energy efficiency of SWIPT-based MEC Network.

In Fig. 4, we analyze the energy efficiency of SWIPT-based MEC Network under different numbers of sub-channels and MEC servers. The results show that no matter how the number of sub-channels and MEC servers change, the energy efficiency of the network will increase with the number of the UE increases. In the case of multiple users, the increase of servers will improve the energy efficiency of the network. In addition, The increase of sub-channels is also beneficial to the network. Therefore, the algorithm can be used in the networks with more sub-channels.

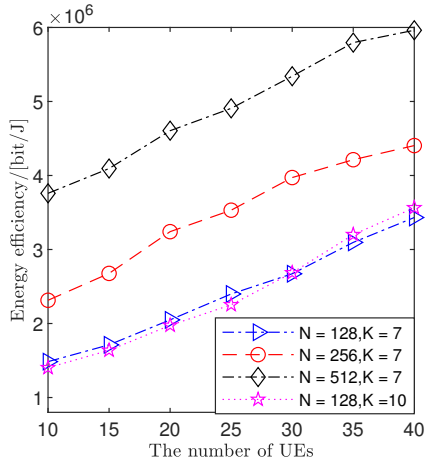

FIGURE 4. Analysis of sub-channels and MEC server in the network.

Fig. 5 shows the relationship between SWIPT-based MEC network energy efficiency and the task lengths. Keeping other parameters unchanged, when the length of tasks reached $2500bit$, the energy efficiency decreases significantly. This is because the maximum computing capability for each MEC server has an threshold. When the computing capacity of the MEC server is exceeded its threshold, other UEs that request for computation offloading will be rejected. So these UEs have to compute their tasks locally. As shown in Fig. 5, with the increase of CPU, the capacity of MEC server is strengthened, this problem will be solved. And this paper assumes that multiple CPUs cooperate perfectly.

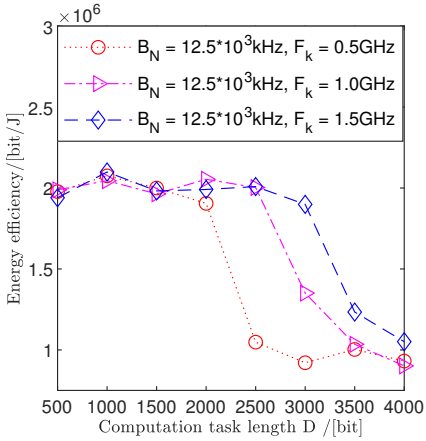

FIGURE 5. Influence of sub-channel bandwidth and computing capability of MEC server on energy efficiency according to different task lengths.

Fig. 6 shows that the energy efficiency of the SWIPT-based MEC network will decrease as the number of EH sub-channels increases. This is because the sub-channel held by ID is used for EH, then the UE has to increase the uplink power in order to meet the rate requirement. Besides, the energy efficiency of the computation offloading model is obeyed $\frac{log_2(1+x)}{x}$, and the energy efficiency of local computing mode is only related to the calculation frequency, so the uplink power of the network is inversely proportional to the energy efficiency.

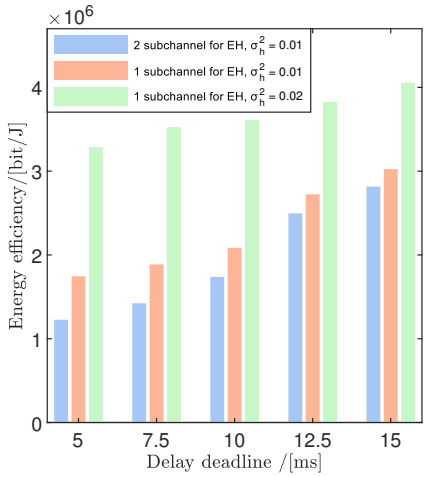

FIGURE 6. Analysis of the network under different delay times and different channel gains.

It is worth mentioning that the error of channel estimation

is inevitable in reality [43]. In order to capture this effect, we try to simulate the reduction of channel estimation error in Fig. 6 to observe the influence of channel quality on the network. We assume that σ_h^2 is the variance of the complex Gaussian distribution ($h \sim \mathcal{CN}(0, \sigma_h^2)$). The results show that the energy efficiency of the model increases as the error decreases. Additionally, as the delay increases, the uploading power of UEs can be decreased. Thereby, the overall energy efficiency of the network can be improved. It also can be seen in Fig. 6, the difference between $\sigma_h^2 = 0.01$ and $\sigma_h^2 = 0.02$ becomes smaller at $12.5ms$. This is because as t^{tolerant} increases, both F^{LC} and P^{UL} decrease. According to (4), (8), (25) and (35), the energy efficiency of local computation is more sensitive to the increase of t^{tolerant}. That makes it exceed the energy efficiency of computation offloading under the condition of $\{t^{\text{tolerant}} = 12.5 \sim 15ms, \sigma_h^2 = 0.01\}$, resulting in abnormal results. It can be seen from Fig. 6 that when the channel estimation error is small, the UE is more willing to offload the task to the MEC server. If the channel estimation error remains the same, the UE tends to offload the calculation task to the MEC server when processing urgent tasks, and tends to calculate the task itself when processing non-urgent tasks.

VI. CONCLUSION

In this paper, we propose an adaptive SWIPT-based MEC network, and jointly optimize the computation offloading strategy and sub-channel allocation. Since the energy efficiency optimization model is a non-convex MINLP problem, we propose an algorithm framework of three-part alternate optimization. The framework divides the optimization variables into three parts, and solves each part in turn. Through alternate optimization, the framework will approach the optimal solution in a spiraling way. The simulation results show that in terms of energy efficiency, the proposed binary offloading bat algorithm is 11% better than the [39] algorithm in solving the problem of sub-channel allocation and computation offloading strategies.

REFERENCES

[1] M. A. Inamdar and H. V. Kumaraswamy, "Energy Efficient 5G Networks: Techniques and Challenges," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020, pp. 1317-1322.

[2] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2018–2023[EB/OL]. Cisco, White Paper, Mar. 2020.

[3] E. Dahlman,S. Parkvall, andJ. Skold. 4G, LTE Evolution and the Road to 5G, Third Edition. Third ed. 2016. Web.

[4] D. Jiang, Z. Wang, Z. Lv and W. Li, "Smart Antenna-based Multi-hop Highly-Energy-Efficient DSA Approach to Drone-assisted Backhaul Networks for 5G," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 2020, pp. 883-887.

[5] F. Busacca, L. Galluccio and S. Palazzo, "Drone-assisted Edge Computing: a game-theoretical approach," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 2020, pp. 671-676.

[6] Q. Zhang, H. Sun, Z. Wei and Z. Feng, "Sensing and Communication Integrated System for Autonomous Driving Vehicles," PIEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 2020, pp. 1278-1279.

[7] A. Dua, A. Dutta, N. Zaman and N. Kumar, "Blockchain-based E-waste Management in 5G Smart Communities," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 2020, pp. 195-200.

[8] P. Jacquet, D. Popescu and B. Mans, "Connecting flying backhauls of drones to enhance vehicular networks with fixed 5G NR infrastructure," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 2020, pp. 472-477.

[9] X. Chen, L. Jiao, W. Li and X. Fu, "Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing," in IEEE/ACM Transactions on Networking, vol. 24, no. 5, pp. 2795-2808, October 2016.

[10] A. Rudenko, P . Reiher, G. J. Popek, and G. H. Kuenning, "Saving portable computer battery power through remote process execution," Journal of ACM SIGMOBILE on Mobile Computing and Communications Review, vol. 2, no. 1, January 1998.

[11] N. Janatian, I. Stupia and L. Vandendorpe, "Optimal Offloading Strategy and Resource Allocation in SWIPT-based Mobile-Edge Computing Networks," 2018 15th International Symposium on Wireless Communication Systems (ISWCS), Lisbon, 2018, pp. 1-6.

[12] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," IEEE Commun. Surveys Tuts., vol. 68, no. 6, pp. 6207–6211, Jun. 2019.

[13] D. Kim, H. Lee, and D. Hong, "A survey of in-band full-duplex transmission: From the perspective of PHY and MAC layers," IEEE Commun.Surveys Tuts., vol. 17, no. 4, pp. 2017–2046, 4th Quart. 2015.

[14] 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.

[15] G. P. Perrucci, F. H. P. Fitzek and J. Widmer, "Survey on Energy Consumption Entities on the Smartphone Platform," 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), Yokohama, 2011, pp. 1-6.

[16] P. K. D. Pramanik et al., "Power Consumption Analysis, Measurement, Management, and Issues: A State-of-the-Art Review of Smartphone Battery and Energy Usage," in IEEE Access, vol. 7, pp. 182113-182172, 2019.

[17] L. R. V arshney, "Transporting information and energy simultaneously," in Proc. IEEE Int. Symp. Inf. Theory, Jul. 2008, pp. 1612–1616.

[18] R. Zhang and C. K. Ho, "MIMO Broadcasting for Simultaneous Wireless Information and Power Transfer," in IEEE Transactions on Wireless Communications, vol. 12, no. 5, pp. 1989-2001, May 2013.

[19] H. Mirghasemi, L. Vandendorpe and M. Ashraf, "Optimal Online Resource Allocation for SWIPT-Based Mobile Edge Computing Systems," 2020 IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Korea (South), 2020, pp. 1-8.

[20] Y. Kim, B. C. Jung, I. Bang and Y. Han, "Adaptive Proportional Fairness Scheduling for SWIPT-Enabled Multicell Downlink Networks," 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 2019, pp. 1-6.

[21] A. I. Akin, H. Mirghasemi and L. Vandendorpe, "SWIPT-based Mobile Edge Computing Systems: A Stochastic Geometry Perspective," 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Istanbul, Turkey, 2019, pp. 1-7.

[22] K. Huang and E. Larsson, "Simultaneous Information and Power Transfer for Broadband Wireless Systems," in IEEE Transactions on Signal Processing, vol. 61, no. 23, pp. 5972-5986, Dec.1, 2013.

[23] J. Fu, J. Hua, J. Wen, H. Chen, W. Lu and J. Li, "Optimization of Energy Consumption in the MEC-Assisted Multi-User FD-SWIPT System," in IEEE Access, vol. 8, pp. 21345-21354, 2020.

[24] Z. Wen, K. Yang, X. Liu, S. Li and J. Zou, "Joint Offloading and Computing Design in Wireless Powered Mobile-Edge Computing Systems With Full-Duplex Relaying," in IEEE Access, vol. 6, pp. 72786-72795, 2018.

[25] D. Kivanc, Guoqing Li and Hui Liu, "Computationally efficient bandwidth allocation and power control for OFDMA," in IEEE Transactions on Wireless Communications, vol. 2, no. 6, pp. 1150-1158, Nov. 2003.

[26] S. Rezvani, N. Mokari and M. R. Javan, "Uplink Throughput Maximization in OFDMA-Based SWIPT Systems with Data Offloading," Electrical Engineering (ICEE), Iranian Conference on, Mashhad, 2018, pp. 572-578.

[27] M. Li, S. Yang, Z. Zhang, J. Ren and G. Yu, "Joint subcarrier and power allocation for OFDMA based mobile edge computing system," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, 2017, pp. 1-6.

[28] S. Yin and Z. Qu, "Resource Allocation in Multiuser OFDM Systems With Wireless Information and Power Transfer," in IEEE Communications Letters, vol. 20, no. 3, pp. 594-597, March 2016.
[29] W. Lu, Y. Gong, J. Wu, H. Peng and J. Hua, "Simultaneous Wireless Information and Power Transfer Based on Joint Subcarrier and Power Allocation in OFDM Systems," in IEEE Access, vol. 5, pp. 2763-2770, 2017.
[30] M. Konstantinos, A. Adamis and P. Constantinou, "Receiver architectures for OFDMA systems with subband carrier allocation," 2008 14th European Wireless Conference, Prague, 2008, pp. 1-7.
[31] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo and D. O. Wu, "Energy-Optimal Mobile Cloud Computing under Stochastic Wireless Channel," in IEEE Transactions on Wireless Communications, vol. 12, no. 9, pp. 4569-4581, Sep. 2013.
[32] Y. Liu and X. Wang, "Information and Energy Cooperation in OFDM Relaying: Protocols and Optimization," in IEEE Transactions on Vehicular Technology, vol. 65, no. 7, pp. 5088-5098, July 2016.
[33] F. Zhou, Y. Wu, R. Q. Hu and Y. Qian, "Computation Efficiency in a Wireless-Powered Mobile Edge Computing Network with NOMA," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, pp. 1-7.
[34] Y. Zhang, J. He and S. Guo, "Energy-Efficient Dynamic Task Offloading for Energy Harvesting Mobile Cloud Computing," 2018 IEEE International Conference on Networking, Architecture and Storage (NAS), Chongqing, 2018, pp. 1-4.
[35] K. T. K. Cheung, S. Yang and L. Hanzo, "Achieving Maximum Energy-Efficiency in Multi-Relay OFDMA Cellular Networks: A Fractional Programming Approach," in IEEE Transactions on Communications, vol. 61, no. 7, pp. 2746-2757, July 2013.
[36] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Found. Trends Mach. Learn., vol. 3, no. 1, pp. 1–122, Jan. 2011.
[37] S. Bi and Y. J. Zhang, "Computation Rate Maximization for Wireless Powered Mobile-Edge Computing With Binary Computation Offloading," in IEEE Transactions on Wireless Communications, vol. 17, no. 6, pp. 4177-4190, June 2018.
[38] Z. Chang, Z. Wang, X. Guo, C. Yang, Z. Han and T. Ristaniemi, "Distributed Resource Allocation for Energy Efficiency in OFDMA Multicell Networks With Wireless Power Transfer," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 2, pp. 345-356, Feb. 2019.
[39] Y. Wu, Y. Wang, F. Zhou and R. Qingyang Hu, "Computation Efficiency Maximization in OFDMA-Based Mobile Edge Computing Networks," in IEEE Communications Letters, vol. 24, no. 1, pp. 159-163, Jan. 2020.
[40] X. Yang, "A new metaheuristic bat-inspired algorithm," In Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Berlin, 2010, pp 65–74.
[41] X. Yang, X. Yu, H. Huang and H. Zhu, "Energy Efficiency Based Joint Computation Offloading and Resource Allocation in Multi-Access MEC Systems," in IEEE Access, vol. 7, pp. 117054-117062, 2019.
[42] F. Chen, J. Fu, Z. Wang, Y. Zhou and W. Qiu, "Joint Communication and Computation Resource Optimization in FD-MEC Cellular Networks," in IEEE Access, vol. 7, pp. 168444-168454, 2019.
[43] D. W. K. Ng and R. Schober, "Cross-Layer Scheduling for OFDMA Amplify-and-Forward Relay Networks," in IEEE Transactions on Vehicular Technology, vol. 59, no. 3, pp. 1443-1458, March 2010.

XUEFEI E was born in Beijing, China in 1997. He received the B.S. degree in Electrical and Information Engineering from the University of Beijing Information Science and Technology University, in June 2019. He is currently pursuing a M.S. degree in Communication Engineering at the University of Science and Technology Beijing. His research interests include intelligent communication and distributed computing.

ZHONGGUI MA received his Ph.D. degree in pattern recognition and intelligent system at Beijing Institute of Technology, China, in 2006. He was a visiting scholar at the University of California, Santa Cruz from August 2014 to September, 2015. Currently, he is a associate professor of School of Computer and Communication Engineering in University of Science and Technology Beijing, China. He has published more than 60 research papers and 10 books. His main research interests include broadband wireless communication and mobile communication.

KAIHANG YU was born in Fuzhou, Jiangxi province, China in 1995. He is currently pursuing a M.S. degree in Communication Engineering at University of Science and Technology Beijing. His current research interests include wireless communication and mobile communication.