IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

SPECIAL SECTION ON DATA MINING FOR INTERNET OF THINGS

Received May 18, 2019, accepted May 28, 2019, date of publication June 5, 2019, date of current version July 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2921114

Joint Offloading and Transmission Power Control for Mobile Edge Computing

JUN LIU, PAN LI, JIANQI LIU, AND JINFENG LAI
School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author: Pan Li (lee79@ieee.org)

This work was supported in part by the National Science Foundation of China under Grant 61701091.

ABSTRACT With the exploding growth of smart devices and application of 5G communications, there are more and more computation-intensive tasks which are delay-sensitive. Mobile edge computing (MEC) is a new paradigm that provides rich computing resources in close proximity to mobile users. However, how to make the decision of whether offloading or not as well as control the transmission power jointly remains a challenge. In this paper, we first study the joint multiuser offloading and transmission power control optimization problem in a multi-channel wireless interference scenario. We formulate the joint optimization problem into a system-wide computation overhead minimization problem as a mixed integer non-linear program. We then propose an efficient semi-distributed algorithm consisting of two subalgorithms of offloading scheme and transmission power control. The numerical simulation results demonstrate that the proposed algorithm can effectively reduce system-wide computation overhead and increase the number of beneficial edge computing users, compared with all local or edge computing and fixed transmission power schemes.

INDEX TERMS Mobile edge computing, computation offloading, transmission power.

I. INTRODUCTION

With the explosive growth of smart devices such as smartphones, laptops, wearable deivices, automotive equipments and etc., there are a broad variety of emerging applications such as image and video processing [1], virtual reality (VR) [2], internet of vehiculars [3], smart city [4] automatic drive, etc. However, the vast majority of these applications and services are generally computational intensive, high energy-consuming and delay sensitive, which can hardly be able to tackle by most of mobile devices today due to their limited computation resources and battery life [5], [6]. So, there is a conflict between the urgent needs for resources and resource limited smart devices, which is a severe challenge for future communication network.

However, a new trend that the powerful Clouds are becoming numerous small cloudlets relatively and moving towards the network edges gradually [7] has emerged in recent years. These cloudlets can provide computing services to process computational intensive and delay sensitive tasks for mobile devices. The paradigm aforementioned is called Mobile Edge Computing (MEC) [8], [9]. Since they are located in close proximity to the mobile devices, the transmission latency between the mobile device and a mobile edge cloud is much lower compared with the remote cloud and the computation capacity is more powerful than mobile deivces though weaker than the remote cloud. So, MEC is a compromise for resource-rich but remote cloud and resource-constrained mobile devices. As a promising solution in the 5G communications, MEC provides ubiquitous computation augmenting services for mobile devices users, which can ease the burden of devices and improve the quality of service (Qos) [8]–[13].

For the moment, it is still difficult to implement an actual MEC system. The key challenge is how to design an efficient computation offloading scheme. To enable benefits from MEC, the following problems shoule be addressed properly: (i) Which tasks of an users should be offloaded onto the edge cloud, and which channel the tasks offload through? (ii) What is the optimal transmission power when the user decide to offload the tasks? However, these two problems are very challenging to solve since the first problem alone is NP-hard as we will show later, let alone the two problems joint together.

In this paper, we study the two problems above in detail and try to design an efficient computation offloading scheme and a power control method for MEC. For computation offloading problem, it is not only an integer programming but vulnerable to oscillations when make the decisions for all the users

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

simultaneously since the user doesn't know the other users decision ahead of time. If making decisions for the users one by one, it is too inefficient. As for power control problem, if the transmission power is too high, it may generate a severe interference to other users in the same channel and be energy inefficient. Otherwise, the transmission rate will be low, which lead to long transmission time, thus losing the meaning of MEC. Not only that, the status of offloading scheme and transmission power interact with each other.

We formulate the joint problem into a mixed integer non-linear program problem and adopt alternative optimization method to solve it. The aim of the joint problem is to minimize the system-wide computation overhead in terms of both communication and computation aspects of local and edge computing. We first transform the joint problem into two sub-problems: a multi-ratio fractional programming sub-problem (i.e., power control problem) and an integer programming sub-problem (i.e., offloading scheme problem). Then, we use alternative optimization techniques to obtain an efficient solution for the joint problem. Specifically, we model the joint problem among multiple mobile device users for MEC in a multi-channel wireless environment [14]. The performance of proposed method is evaluated and compared with local computing by all users scheme, edge computing by all users scheme and fixed transmission power method. The main contributions of this paper include:

- We formulate the joint problem into a mixed integer non-linear program problem, which takes into account two aspects of both communication and computation in local and edge computing, then transform it into two sub-problems. On the basis of analysis, power control problem is a multi-ratio fractional programming problem with coupled optimization variables. We adopt quadratic transform [15] and interference pricing mechanism [16], [17] to solve the sub-problem in a distributed way, which can help to ease the heavy burden of the complex centralized management by the cloud operator [14]. As for offloading scheme problem, it is a integer programming, which is NP-hard as we will see later. We first transform it into a 0-1 programming. Then similar with the work in [18], [19], we use a series of re-weighted linear functions to replace the integer variables, but we modify the algorithm in order to avoid the oscillation in the process of solving.
- We design an efficient algorithm to solve the problem, which reduce the system-wide computation overhead significantly. To be specific, when fixing the offloading decision, we optimize the transmission power in a distributed way. Based on the solution above, we adopt offloading scheme algorithm to give a effective offloading decision. Numerical results demonstrate that the proposed algorithm can achieve excellent offloading performance.

The rest of this paper is organized as follows. We first review the related work in Section II. Section III describes the system model and joint optimization problem formulation. Then, we propose the distributed power control algorithm and offloading scheme algorithm in Section IV. The numerical results and analysises are given in Section V. Finally, we conclude in Section VI.

II. RELATED WORKS

There have been a variety of offloading schemes proposed in the literature. Here, We only focus on multiuser MEC systems. In mobile edge computing, the offloading scheme and radio-and-computational resources are two hot research points. Xu *et al.* [14] designed a game theoretic approach for the computation offloading decision making problem among multiple mobile device users for mobile-edge cloud computing. Chen *et al.* [19] presented an algorithm to jointly optimize the offloading decisions and the allocation of computing resource in order to minimize the task duration in the software defined ultra dense network. Chen *et al.* in [20] aimed to jointly optimize the offloading decisions of all users' tasks as well as the allocation of computation and communication resources, to minimize the overall cost of energy, computation, and delay for all users. Lyu *et al.* [21] proposed a heuristic offloading decision algorithm, which jointly optimizes the offloading decision and computation resources to maximize system utility. Sardellitti *et al.* [22] studied the offloading problem as the joint optimization of the radio resource and the computational resources in order to minimize the overall users' energy consumption, and they designed an iterative algorithm based on a novel successive convex approximation technique to solve the joint proposed. Wang *et al.* [23] considered a wireless powered multiuser MEC system, where an access point play the roles of power source and an MEC server simultaneously, and the authors develop an innovation framework by jointly optimizing the energy transmit beamforming at access point as well as the time allocation among users. Tran and Pompili [24] studied the problem of joint task offloading and resource allocation in order to maximize the users' task offloading gain, which is measured by a weighted sum of reductions in task completion time and energy consumption. However, these works mainly considered the task offloading problem with a specific resource allocation problem jointly in an MEC system.

Furthermore, a few existing works considered both task offloading and transmission power control in MEC, Guo *et al.* [25] developed a distributed energy-efficient dynamic offloading and resource scheduling algorithm consisting of three subalgorithms of computation offloading selection, clock frequency control, and transmission power control. Nevertheless, they only studied the scenario that there was no interference between the users who offload onto the edge cloud, and the optimization problem becomes convex when relax the binary computation offloading decision variable, therefore our problem is more difficult to solve directly. In this paper, we consider a more practical scenario that the wireless channel is limited and each user may suffer

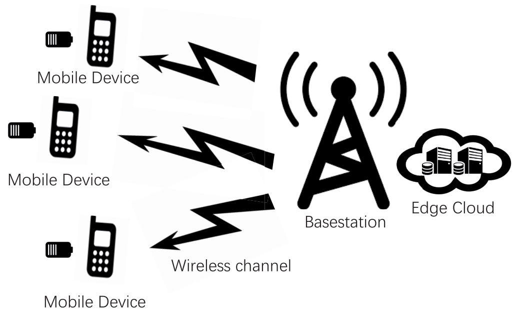

FIGURE 1. Illustration of mobile edge computing with multiple device users in a multi-channel wireless environment.

interference from others users in the same channel [14] and propose an efficient offloading scheme as well as transmission power control method jointly.

III. SYSTEM MODEL

In this section, we first introduce system model as illustraed in Fig. 1. We consider a celluar uplink communication system which consists of a wireless base-stastion b equipped with edge clouds and N mobile users distributed uniformly in the celluar, denoted by $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$. We assume that each user in $\mathcal{U}$ has a computation-intensive task to be solved locally or offload to the nearby edge cloud totally, which means that we consider a binary offloading problem in this paper. Similar to many previous studies, we describe user u_n's computation task as $\mathcal{T}_n = (s_n, c_n)$, where s_n stands for the size of computation task $\mathcal{T}_n$ (e.g., the input data and program code) to be offload toward the edge cloud, and c_n stands for the number of CPU cylces needed in total to fulfil the computation task $\mathcal{T}_n$. Since the states of communication and computation are the two main infulence factors in making the decision on whether offload or not, in the following, we introduce the comminication and computation models in details.

A. COMMUNICATION MODEL

We first introduce communication model for wireless uplink channel in a celluar. In this paper, we consider there are M orthogonal subchannels which are donoted as $\mathcal{SC} = \{1, 2, \ldots, M\}$. Let $h_{n,b}$ donotes the channel gain between the mobile device user u_n and the base-station b. Similar to many previous studies such as in MEC [6], [19], [24], [25], in order to simplify the model complexity, we assume quasi-static channels where the channel gain $h_{n,b}$ remains constant during each offloading period. Furthermore, denote $d_n \in \{0\} \cup \mathcal{SC}$ as the computation offloading decision variable of the mobile device user u_n. To be more specific, $d_n \neq 0$ means that the mobile user u_n decides to offload the comuptation task through the corresponding wireless channel d_n, while $d_n = 0$ means that the user chooses to compute tasks locally. Denote transmission power of the user u_n as p_n, the transmission power vector as $\boldsymbol{p} = (p_1, p_2, \ldots, p_N)$, and the decision vector as $\boldsymbol{d} = (d_1, d_2, \ldots, d_N)$.Then the uplink data rate of user u_n which offload decision variable $d_n > 0$ can be expressed as

$$r_n(\boldsymbol{d}, \boldsymbol{p}) = B \log_2 \left(1 + \frac{p_n h_{n,b}}{\sigma^2 + \sum_{i \in \mathcal{U} \setminus \{n\} : d_i = d_n} p_i h_{i,b}} \right), \quad (1)$$

where B is the bandwidth of each subchannel, and σ^2 denotes the background noise power. It is worth noting that we foucus on soloving the computation task offloading problem under the wireless interference channel in a distributed way, and consider certain channel access scheme such as CDMA that can enable multiple users to share the same spectrum resource simultaneously.

From (1), we can obtain the transmission delay of user u_n for offloading the task $\mathcal{T}_n$ as follows:

$$t_n^C(\boldsymbol{d}, \boldsymbol{p}) = \frac{s_n}{r_n(\boldsymbol{d}, \boldsymbol{p})}. \quad (2)$$

Further we can obtain the transmission energy comsumption of user u_n as follows:

$$\varepsilon_n^C(\boldsymbol{d}, \boldsymbol{p}) = \frac{p_n s_n}{r_n(\boldsymbol{d}, \boldsymbol{p})}. \quad (3)$$

From (1), (2) and (3), we can find that if a user u_n intends to have a lower transmission delay, it needs to get a faster uplink data rate $r_n(\boldsymbol{d}, \boldsymbol{p})$, implying a higher transmission power p_n. However, if p_n is too large, the user u_n will cause more serious interference to other users on the same wireless acess channel during the computation offolading period, leading to low uplink date rate of other users, or even making other users change their decisions into computing locally, which would decrease the efficiency of the overall system. Therefore, we need to find proper transimission powers of all users to make the overall system more efficient.

B. COMPUTATION MODEL

A user can choose to accomplish its computation task locally or offload it to the edge cloud according to its communication and computation conditions. In the following we discuss the computation model for both local computing and MEC.

1) LOCAL COMPUTING

For local computing, we define f_n^L as the CPU computing capability of mobile device user u_n, and different mobile devices may have different CPU computing capabilities. Thus, the execution latency for task $\mathcal{T}_n$ can be given by

$$t_n^L = \frac{c_n}{f_n^L}. \quad (4)$$

Besides, computation energy comsumption is another important measurement for local computing. It is known from [26] and [27] that the CPU's clock frequency is approximately linear proportianal to the volatge supply when operating at low voltage limits. So the energy consumption for a CPU cycle can be given by $\kappa (f_n^L)^2$, where κ is a coefficient related to the hardware architecture. Based on the aforementioned result, the energy consumption for local computing can be derived by:

$$\varepsilon_n^L = \kappa (f_n^L)^2 c_n. \tag{5}$$

To simplify the notation, we define $\rho_n = \kappa (f_n^L)^2$ for user u_n. Thus (5) can be rewritten as $\varepsilon_n^L = \rho_n c_n$.

From (4) and (5), we can observe that if a user u_n intends to have a lower execution latency, a higher CPU clock frequency f_n^l is needed, thus consuming more execution energy. In practice, the mobile devices have limited CPU clock frequency and hence are unable to solve the computation-intensive task, or else the energy consumption would be very high and drains the mobile devices batteries very quickly. Thus, it is usually more efficient to offload computation-intensive tasks to the edge cloud.

Note that similar to previous works [19], [28], we ignore the execution latency and energy consumption of other hardware components (e.g., RAM) in this paper, which are generally much less significant.

2) MOBILE EDGE COMPUTING

For the mobile edge computing, let f_n^C denote the allocated computing resource (CPU-cycle frequency) for the mobile user u_n. Note that here we assume that all the users can access the edge cloud services, and that the edge cloud has enough computing resources to support all users' offloading requests. Thus, the edge cloud execution time can be calculated as

$$t_{n,exe}^C = \frac{c_n}{f_n^C}. \tag{6}$$

Similar to that in many previous studies like [14], [29], [30], we ignore the transmission delay for the edge cloud to send the computation result back to the user, since the size of the computation result is often much smaller than the size of offloaded data.

3) SYSTEM OVERHEAD

Based on models above, we compute the local computing and MEC overhead in the following. For local computing, the overhead of the user u_n,, denoted by H_n^L, including execution latency and energy consumption can be obtained as

$$H_n^L = \alpha_n^t t_n^L + \alpha_n^e \varepsilon_n^L, \tag{7}$$

where $\alpha_n^t, \alpha_n^e \in [0, 1]$ denote the weighting coefficients of execution latency and energy consumption, respectively [14]. When the user u_n's battery has sufficient power and mainly care about the execution latency, it can set $\alpha_n^t = 1$ and $\alpha_n^e = 0$. When the user u_n is concerned about the two factors equally, it can set $\alpha_n^t = 0.5$ and $\alpha_n^e = 0.5$. Obviously, the two coefficients have great impact on the decision of whether to offload or not.

For edge cloud computing, the overhead of the user u_n, denoted by H_n^C, including transmission delay, execution latency and energy consumption can be obtained as

$$H_n^C = \alpha_n^t (t_n^C(\boldsymbol{d}, \boldsymbol{p}) + t_{n,exe}^C) + \alpha_n^e \varepsilon_n^C(\boldsymbol{d}, \boldsymbol{p}). \tag{8}$$

Based on the system overhead model above, we will formulate an optimization problem in terms of transmission power and offloading scheme for MEC in the next section.

IV. JOINT OFFLOADING AND TRANSMISSION POWER CONTROL

In this section, we formulate a problem which considers joint offloading and transmission power control. Our objective in this papaer is to minimize the system-wide overhead. Specifically, we define a binary variable $\psi_n \in \{0, 1\}$ to indicate that the task $\mathcal{T}_n$ is solved locally ($\psi_n = 1$) or in the edge cloud ($\psi_n = 0$). We also define an indicator vector $\Psi = (\psi_1, \psi_2, \ldots, \psi_n)$. Thus, the joint offloading scheme and power control problem can be formulated as follows:

$$\begin{aligned}
\underset{\Psi, \boldsymbol{d}, \boldsymbol{p}}{\text{minimize}} \quad & \sum_{i=1}^{N} [\psi_i H_i^L + (1 - \psi_i) H_i^C] && \text{(9a)}\\
\text{subject to} \quad & \psi_i \in \{0, 1\}, \quad \forall i = 1, \ldots, N. && \text{(9b)}\\
& d_i = 0, \forall \psi_i = 0. && \text{(9c)}\\
& d_i \in \{0\} \cup \mathcal{SC}, \quad \forall i = 1, \ldots, N. && \text{(9d)}\\
& 0 \le p_i \le p_{\max}, \quad \forall i \in \{l | d_l > 0\}. && \text{(9e)}\\
& p_i = 0, \quad \forall \psi_i = 1. && \text{(9f)}
\end{aligned}$$

where $p_{\max}$ denotes the transmission power constraint. This joint optimization problem is a mixed-integer nonlinear programming problem, and hence generally NP-hard and very challenging to solve [31]. In what follows, we utilize alternative optimization method to solve the joint problem.

Specifically, we design an algorithm that minimize the overall system overhead for MEC in terms of the offloading scheme $(\Psi, \boldsymbol{d})$ and the transmission power vector $\boldsymbol{p}$. First, let $\mathbb{M}$ and $\mathbb{P}$ be the feasible sets for $(\Psi, \boldsymbol{d})$ and $\boldsymbol{p}$ respectively. Then, we expand and define the objective function in (9) as

$$\begin{aligned}
f(\Psi, \boldsymbol{d}, \boldsymbol{p}) = \sum_{i=1}^{N} [\psi_i (\frac{\alpha_i^t c_i}{f_i^l} + \alpha_i^e \rho_i c_i) + (1 - \psi_i) \\
\times (\alpha_i^t (\frac{s_i}{r_i(\boldsymbol{d}, \boldsymbol{p})} + \frac{c_i}{f_i^C}) + \frac{\alpha_i^e p_i s_i}{r_i(\boldsymbol{d}, \boldsymbol{p})})]. \quad (10)
\end{aligned}$$

As illustrated in Fig. 2, we utilize the alternative optimization method to solve this joint problem by decomposing it into two sub-problems:

- Power control: Given a fixed decision tuple $(\Psi^0, \boldsymbol{d}^0) \in \mathbb{M}$, the origin problem in (9) transforms into a multi-ratio fractional programming problem with coupling optimization variables. We adopt quadratic transform [15] and interference pricing mechanism [16], [17] to sovle the problem in a distributed way. The optimal solution can be denoted as $f(\Psi^0, \boldsymbol{d}^0, \boldsymbol{p}^*)$.
- Offloading scheme: Based on the solution $\boldsymbol{p}^* \in \mathbb{P}$ above, the original optimization problem in (9) degrades into an integer programming problem which is denoted by $f(\Psi, \boldsymbol{d}, \boldsymbol{p}^*)$. Similar to the work in [18], [19], we use a

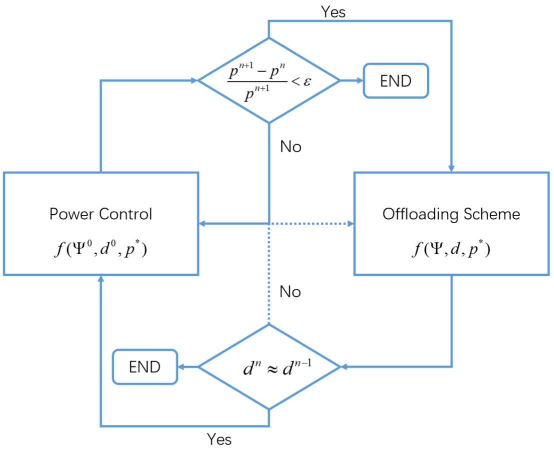

FIGURE 2. **Illustration of the overall optimization algorithm.**

linear function to substitute the decision variable, but we improve the algorithm in order to have a higher convergence rate.

Finally, we will prove the convergence of the proposed algorithm.

A. POWER CONTROL

Since only when a user decides to offload its task to the edge cloud, it needs to find an optimal transmission power. We denote a set $\mathcal{O} = \{i|\psi_i = 0, i \in [1, N]\}$, given the decision tuple $(\Psi^0, \boldsymbol{d}^0) \in \mathbb{M}$. Then the objective function (9a) can be rewritten as follows:

$$f(\boldsymbol{d}^0,\boldsymbol{p}) = \sum_{i\in\mathcal{O}}[(\alpha_i^t(\frac{s_i}{r_i(\boldsymbol{d}^0,\boldsymbol{p})}+\frac{c_i}{f_i^C})+\frac{\alpha_i^e p_i s_i}{r_i(\boldsymbol{d}^0,\boldsymbol{p})})]. \tag{11}$$

among which the term $\frac{c_i}{f_i^C}$ is uncorrelated to the optimization variables $\boldsymbol{p}$. So (11) can be further simplified as

$$f(\boldsymbol{d}^0,\boldsymbol{p}) = \sum_{i\in\mathcal{O}}\frac{(\alpha_i^t+\alpha_i^e p_i)s_i}{B\log_2\left(1+\frac{p_i h_{i,b}}{\sigma^2+\sum_{k\in\mathcal{U}\setminus\{i\}:d_k=d_i}p_k h_{k,b}}\right)}. \tag{12}$$

From (12), it is easy to see that the problem is not only a multi-ratio fractional programming, but also the optimization variables respected to $\boldsymbol{p}$ are coupled with each other. Due to the fact that only when the users offload the tasks through the same wireless uplink channel, they have channel interference with each other. Let $\boldsymbol{p}_m \triangleq \{p_k|k \in \mathcal{O} : d_k = d_m\}$, then (12) can be decomposed into follows according to different subchannels:

$$f(\boldsymbol{d}^0,\boldsymbol{p}) = \sum_{m=1}^{M}\sum_{j\in\mathcal{O}:d_j=d_m}\frac{(\alpha_j^t+\alpha_j^e p_j)s_j}{r_j(\boldsymbol{d}^0,\boldsymbol{p}_m)}. \tag{13}$$

Since there is no interference among different subchannels, we can only focus on one subchannel m ($m \in \{1,\ldots,M\}$), and the optimal power control on other subchannels are simply the same. Thus, we have

$$f_m(\boldsymbol{d}^0,\boldsymbol{p}_m) = \sum_{j\in\mathcal{O}:d_j=m}\frac{(\alpha_j^t+\alpha_j^e p_j)s_j}{B\log_2\left(1+\frac{p_j h_{j,b}}{\sigma^2+\sum_{k\in\mathcal{O}\setminus\{j\}:d_k=d_j}p_k h_{k,b}}\right)}. \tag{14}$$

Therefore, the problem that aims at optimizing the transmission power on channel m can be formulated as

$$\underset{\boldsymbol{p}_m}{\text{maximize}} \quad g_m(\boldsymbol{d}^0,\boldsymbol{p}_m) = \sum_{j\in\mathcal{O}:d_j=m} -\frac{(\alpha_j^t+\alpha_j^e p_j)s_j}{r_j(\boldsymbol{d}^0,\boldsymbol{p}_j)} \tag{15a}$$

$$\text{subject to} \quad 0 \le p_j \le p_{\max}, \quad \forall j \in \mathcal{O} : d_j = m. \tag{15b}$$

We can find that (15) is still a multi-ratio fractional programming with coupled optimization variables. In this paper, we adopt quadratic transform [15] to sovle this problem. In particular, the problem (15) is equivalent to

$$\underset{\boldsymbol{p}_m,\boldsymbol{y}}{\text{maximize}} \quad g_m(\boldsymbol{d}^0,\boldsymbol{p}_m,\boldsymbol{y}) = \sum_{j\in\mathcal{O}:d_j=m}(y_j^2 r_j(\boldsymbol{d}^0,\boldsymbol{p}_m) - 2y_j\sqrt{(\alpha_j^t+\alpha_j^e p_j)s_j}) \tag{16a}$$

$$\text{subject to} \quad 0 \le p_j \le p_{\max}, \quad \forall j \in \mathcal{O} : d_j = m. \tag{16b}$$

$$y_j \in \mathbb{R}, \tag{16c}$$

where $\mathbb{R}$ represents the set of real numbers and $\boldsymbol{y}$, which is introduced by the quadratic transform for each user u_j on subchannel m, refers to a collection of variables $\{y_j|j \in \mathcal{U} : d_j = d_m\}$.

Following the quadratic transform algorithm, we optimize $\boldsymbol{p}_m$ and $\boldsymbol{y}$ in an iterative fashion. First, let $z(p_j) = \sqrt{(\alpha_j^t + \alpha_j^e p_j)s_j}$. Thus optimal y_j given fixed $\boldsymbol{p}_m$ is calculated as:

$$y_j^\star = \frac{z(p_j)}{r_j(\boldsymbol{d}^0,\boldsymbol{p}_m)} \tag{17}$$

Second, given fixed $\{y_j^*|j \in \mathcal{O}\}$, we need to find the optimal $\boldsymbol{p}_m$, which, however, is still difficult to be solved yet due to the coupling of optimization variables in the objective function (16a). In the following, we develop a distributed algorithm to solve this optimization problem. Particularly, we employ the interference pricing mechanism, which is an effective algorithm to solve coupling optimization problem, to decompose the optimization problem into a series of independent subproblems.

Recall that $\boldsymbol{p}_m \triangleq \{p_j|j \in \mathcal{O} : d_j = m\}$ and define $\boldsymbol{p}_m^{-j} \triangleq \boldsymbol{p}_m \backslash p_j$ as a set of transmission power on subchannel m of all users except the jth user. Since $r_j(\boldsymbol{d}^0,\boldsymbol{p}_m)$ is a convex function with respect to $\boldsymbol{p}_m^{-j}$ [32]. $r_j(\boldsymbol{d}^0,\boldsymbol{p}_m)$ can be approximated by its first-order Taylor expansion. Let $\boldsymbol{p}_m^{-j(n)}$ denote the optimal solution at the n-th iteration. Then $r_j(\boldsymbol{d}^0,\boldsymbol{p}_m)$

can be approximated by its first-order Taylor expansion at $\boldsymbol{p}_m^{-j(n)}$ as

$$r_j(\boldsymbol{d}^0, \boldsymbol{p}_m) \approx r_j(\boldsymbol{d}^0, p_j, \boldsymbol{p}_m^{-j(n)}) + \sum_{k \in \mathcal{O}\setminus\{j\}:d_k=m} \left\langle \frac{\partial r_j}{\partial p_k}, p_k - p_k^{(n)} \right\rangle, \tag{18}$$

where

$$\left\langle \frac{\partial r_j}{\partial p_k}, p_k - p_k^{(n)} \right\rangle = \left\langle \frac{\partial r_j}{\partial \varphi_j}\frac{\partial \varphi_j}{\partial p_k}, p_k - p_k^{(n)} \right\rangle = -V_j h_{k,b}(p_k - p_k^{(n)}), \tag{19}$$

where φ_j denotes user j's covariance of the background noise and interference from other users, i.e.,

$$\varphi_j = \sigma^2 + \sum_{k \in \mathcal{O}\setminus\{j\}:d_k=m} p_k h_{k,b}, \tag{20}$$

and

$$V_j = -\frac{\partial r_j}{\partial \varphi_j} = B(\varphi_j^{-1} - (\varphi_j + p_j h_{j,b})^{-1}). \tag{21}$$

Based on (18) and (19), the objective function in (16) can be approximated as follows:

$$\begin{aligned} g_m(\boldsymbol{d}^0, \boldsymbol{p}_m, \boldsymbol{y}) \approx & \sum_{j \in \mathcal{O}:d_j=m} \Bigg(y_j^2 \Big(r_j(\boldsymbol{d}^0, p_j, \boldsymbol{p}_m^{-j(n)}) \\ & - \sum_{k \in \mathcal{O}\setminus\{j\}:d_k=m} V_j h_{k,b}(p_k - p_k^{(n)}) \Big) - 2y_j z(p_j) \Bigg) \\ = & \sum_{j \in \mathcal{O}:d_j=m} \Bigg(y_j^2 r_j(\boldsymbol{d}^0, p_j, \boldsymbol{p}_m^{-j(n)}) \\ & - \sum_{k \in \mathcal{O}\setminus\{j\}:d_k=m} y_k^2 V_k h_{j,b}(p_j - p_j^{(n)}) - 2y_j z(p_j) \Bigg) \end{aligned} \tag{22}$$

According to (22), we can find that the approximated version of objective function in is decoupled, so the original problem (16) can be decomposed into a series of subproblems, which can be solved locally. When some constant items are removed, the subproblem can be formulated as

$$\begin{aligned} \underset{p_j, \boldsymbol{y}}{\text{maximize}} \quad & y_j^2 r_j(\boldsymbol{d}^0, p_j, \boldsymbol{p}_m^{-j(n)}) - 2y_j z(p_j) \\ & - \sum_{k \in \mathcal{O}\setminus\{j\}:d_k=m} y_k^2 V_k h_{j,b} p_j & (23a) \\ \text{subject to} \quad & 0 \le p_j \le p_{\max}, \quad \forall j \in \mathcal{O} : d_j = m. & (23b) \\ & y_j \in \mathbb{R}. & (23c) \end{aligned}$$

V_k can be viewed as a interference price, which is other user's payment per unit interference increase to user k. Thus, $V_k h_{j,s} p_m^j$ is user j's payoff for user k due to the interference increase to user k. The third item in (23a) is used to control the interference user j creates to other users.

From (23), we can find that the first and second items are concave on $\mathbb{R}_{++}$ [32], which satisfy the condition of D.C. (difference of two convex/concave functions) form, and the third term is linear. Furthermore, the D.C. programming can be solved by the successive convex approximition (SCA) algorithm. Given $p_j^{(n)}$, the second item can be linearzed as [33]

$$\begin{aligned} 2y_j z(p_j) \approx & \, 2y_j \sqrt{(\alpha_j^t + \alpha_j^e p_j^{(n)}) s_j} \\ & + \frac{y_j \alpha_j^e s_j}{\sqrt{\alpha_j^t + \alpha_j^e p_j^{(n)}) s_j}} (p_j - p_j^{(n)}) \\ \equiv & \, 2y_j \tilde{z}(p_j, p_j^{(n)}). \end{aligned} \tag{24}$$

When remove some constant items from (24), we obtain an approximated version of problem (23) as

$$\begin{aligned} \underset{p_j, \boldsymbol{y}}{\text{maximize}} \quad & y_j^2 r_j(\boldsymbol{d}^0, p_j, \boldsymbol{p}_m^{-j(n)}) - \frac{y_j \alpha_j^e s_j}{\sqrt{\alpha_j^t + \alpha_j^e p_j^{(n)}) s_j}} p_j \\ & - \sum_{k \in \mathcal{O}\setminus\{j\}:d_k=m} y_k^2 V_k h_{j,b} p_j & (25a) \\ \text{subject to} \quad & 0 \le p_j \le p_{\max}, \quad \forall j \in \mathcal{O} : d_j = m. & (25b) \\ & y_j \in \mathbb{R}. & (25c) \end{aligned}$$

Problem (25) is a convex problem, which can be solved by the existing software packages.

The convergence analyses in [16] and [34] show that interference pricing mechanism converges to a KKT point. We will give a simple proof about the convergence of solving the subproblem (25) iteratively. Let $u(p_j) = \sum_{k \in \mathcal{O}\setminus\{j\}:d_k=m} y_k^2 V_k h_{j,b} p_j$, and $z(p_j)$ is a concave function,

$$z(p_j^{(n+1)}) \le \tilde{z}(p_j^{(n+1)}, p_j^{(n)}), \tag{26}$$

and

$$\begin{aligned} & y_j^2 r_j(\boldsymbol{d}^0, p_j^{(n+1)}, \boldsymbol{p}_m^{-j(n)}) - 2y_j z(p_j^{(n+1)}) - u(p_j^{(n+1)}) \\ & \ge y_j^2 r_j(\boldsymbol{d}^0, p_j^{(n+1)}, \boldsymbol{p}_m^{-j(n)}) - 2y_j \tilde{z} p_j^{(n+1)}, p_j^{(n)}) - u(p_j^{(n+1)}). \end{aligned} \tag{27}$$

At the $(n+1)$-th iteration, we assume $p_j^{(n+1)}$ is the optimal solution of problem (25), then,

$$\begin{aligned} & y_j^2 r_j(\boldsymbol{d}^0, p_j^{(n+1)}, \boldsymbol{p}_m^{-j(n)}) - 2y_j \tilde{z}(p_j^{(n+1)}, p_j^{(n)}) - u(p_j^{(n+1)}) \\ & \ge y_j^2 r_j(\boldsymbol{d}^0, p_j^{(n)}, \boldsymbol{p}_m^{-j(n)}) - 2y_j \tilde{z}(p_j^{(n)}, p_j^{(n)}) - u(p_j^{(n)}) \\ & = y_j^2 r_j(\boldsymbol{d}^0, p_j^{(n)}, \boldsymbol{p}_m^{-j(n)}) - 2y_j z p_j^{(n)}) - u(p_j^{(n)}). \end{aligned} \tag{28}$$

From (26) and (27), we can obtain that

$$\begin{aligned} & y_j^2 r_j(\boldsymbol{d}^0, p_j^{(n+1)}, \boldsymbol{p}_m^{-j(n)}) - 2y_j z(p_j^{(n+1)}) - u(p_j^{(n+1)}) \\ & \ge y_j^2 r_j(\boldsymbol{d}^0, p_j^{(n)}, \boldsymbol{p}_m^{-j(n)}) - 2y_j z(p_j^{(n)}) - u(p_j^{(n)}). \end{aligned} \tag{29}$$

From (29), we can find that the value of the original objective function at the $n+1$ items is bigger than that of the iteration n, thus the algorithm is proved to be convergent.

The power control problem is a multi-ratio fractional programming problem with coupling optimization variables. Firstly, we decompose the objective function into several independent parts according to its subchannel. Then, we adopt quadratic transform to decouple the numerator and

Algorithm 1 Distributed Power Control Algorithm

```
Initialization:
Initialize $\boldsymbol{p}$ to a feasible value.
Give an initia offloading decision vector $\boldsymbol{d}^0$.
Set $n = 0$.
end initialization

repeat For each subchannel $m$ in parallel:
    repeat
        Compute the interference prices by (21) and broadcast them to other users.
        Update $\boldsymbol{y}_m$ by (17).
        repeat Set $n = n + 1$, for all users in subchannel $m$:
            Each user solves the problem (25) locally and obtains a optimal solution $\hat{p}_j^{(n+1)}$.
            Update $p_m^{j(n+1)}$ according to
                $$p_j^{(n+1)} = (1-\gamma^{n+1}) + \gamma^{n+1}\hat{p}_j^{(n+1)}.$$
        until $\left|p_j^{(n+1)} - p_j^{(n)}\right| / p_j^{(n+1)} < \epsilon$
    until Convergence
until All the subchannels are convergence
```

the denominator of each ratio term by introducing some suitable auxiliary variables. We optimize the auxiliary variables and the transmission power in a iteration fashion. But still, there are coupled optimization variables in the objective function, interference pricing mechanism is used to decouple the variables and the original problem is first decomposed into a series of subproblems, which can be solved in a distributed way. Finally, we find that the objective function satisfies the D.C. form, and the SCA algorithm is adopted to solve the problem. So, there are two iterations nested with each other. At first iteration, we optimize the auxiliary variables and the transmission power. At second iteration, we optimize the transmission power in a distributed way. To be specific, each user calculates and broadcasts its interference price to other users in the same subchannel. When the user receives the interference prices of all other users in the same subchannel, it update its transmission power by solving the subproblem. Given the convergence threshold ϵ and learning rate $\boldsymbol{\gamma}$, the entire algorithm for power control is summarized in Algorithm 1.

B. OFFLOADING SCHEME PROBLEM

Given the decision tuple $(\Psi^0, \boldsymbol{d}^0) \in \mathbb{M}$, we can obtain the optimal trasmission power solution in (25). Based on the solution $\boldsymbol{p}^* \in \mathbb{P}$ above, the original problem in (9) degrades into an integer programming problem as follows:

$$\underset{\Psi,\boldsymbol{d},\boldsymbol{p}}{\text{minimize}} \quad f(\Psi, \boldsymbol{d}, \boldsymbol{p}^*) \tag{30a}$$

$$\text{subject to} \quad \psi_i \in \{0, 1\}, \quad \forall i = 1, \ldots, N. \tag{30b}$$

$$d_i \in \{0\} \cup \mathcal{SC}, \quad \forall i = 1, \ldots, N. \tag{30c}$$

Similar to the work in [19], we rewrite the optimization problem above by matrixing method to transfer it into a 0-1 integer programming, but [19] has done something wrong with its definition of some matirxes and the inner matrix dimensions do not agree in the formulated optimization problem. We also do some modifications to the algorithm in order to decrease the oscillation in the process of optimization.

Let $\boldsymbol{\Omega} = (\omega_{ij})_{N\times(M+N)}, \omega_{ij} \in \{0, 1\}$ denote the matrixing tuple $(\Psi, \boldsymbol{d}) \in \mathbb{M}$. To be specific, if $\omega_{ij} = 1$ and $j \leq M$, it means the task will be offloaded to the edge cloud through subchannel SC_j of the user u_i, which can be donoted as $\boldsymbol{\Omega}_1$. If $\omega_{ij} = 1, j > M$ and $i + M = j$, it means the task will be processed locally of user u_i, which can be denoted as $\boldsymbol{\Omega}_2$ and it is a diagonal matrix. So, $\boldsymbol{\Omega} = [\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2]$. Further, we rewrite computation task $\mathcal{T}$ and weighting coefficients as follows:

$$\boldsymbol{S} = (s_1, s_2, \ldots, s_N), \tag{31}$$

$$\boldsymbol{C} = (c_1, c_2, \ldots, c_N), \tag{32}$$

$$\boldsymbol{\alpha^t} = (\alpha_1^t, \alpha_2^t, \ldots, \alpha_N^t), \tag{33}$$

$$\boldsymbol{\alpha^e} = (\alpha_1^e, \alpha_2^e, \ldots, \alpha_N^e), \tag{34}$$

$$\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_N), \tag{35}$$

and form two diagonal matrixes to denote local and edge computing resource respectively:

$$\boldsymbol{F}_1 = \begin{bmatrix} \frac{1}{f_1^C} & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{f_2^C} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{f_{N-1}^C} & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{f_N^C} \end{bmatrix}, \tag{36}$$

$$\boldsymbol{F}_2 = \begin{bmatrix} \frac{1}{f_1^L} & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{f_2^L} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{f_{N-1}^L} & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{f_N^L} \end{bmatrix}, \tag{37}$$

and the reciprocals of uplink data rate can form a matrix as

$$\boldsymbol{R} = \begin{bmatrix} \frac{1}{r_{11}} & \cdots & \frac{1}{r_{1M}} \\ \frac{1}{r_{21}} & \cdots & \frac{1}{r_{2M}} \\ \vdots & \vdots & \vdots \\ \frac{1}{r_{N1}} & \cdots & \frac{1}{r_{NM}} \end{bmatrix}_{N\times M}. \tag{38}$$

Let $\circ$ denote the Hadamard product, then the optimization problem in (30) can be rewritten as follows:

$$\begin{aligned} \underset{\boldsymbol{\Omega}}{\text{minimize}} \quad & |\boldsymbol{\alpha}_t \circ \boldsymbol{C}\boldsymbol{F}_1\boldsymbol{\Omega}_1| + |\boldsymbol{\alpha}_t \circ \boldsymbol{C}\boldsymbol{F}_2\boldsymbol{\Omega}_2| \\ & + |\boldsymbol{\alpha}_e \circ \boldsymbol{\rho} \circ \boldsymbol{C}\boldsymbol{\Omega}_2| + |\boldsymbol{\alpha}_t \circ \boldsymbol{S} \circ \boldsymbol{R} \circ \boldsymbol{\Omega}_1| \\ & + \left|\boldsymbol{\alpha}_e \circ \boldsymbol{p}^* \circ \boldsymbol{R} \circ \boldsymbol{\Omega}_1\right| && \text{(39a)} \\ \text{subject to} \quad & \omega_{ij} \in \{0, 1\}\,. && \text{(39b)} \\ & \sum_{j=1}^{M+N} \omega_{ij} = 1, \quad i = 1, \ldots, N. && \text{(39c)} \end{aligned}$$

The objective function in (39) with respect to $\boldsymbol{\Omega}$ aims to reduce the system-wide computation overhead. The constraints mean that the task can only process at one place. Obviously, the problem is a non-convex optimization problem. Specifically, it is a combinatorial optimization over a multi-dimensional descrete space (i.e., $\{0, 1, \ldots M\}^N$), which is NP-hard.

For the purpose of solving the problem, we first define a variable which is similar to Ω,

$$\lambda_{ij} \geq 0, \quad \sum_{j=1}^{M+N} \lambda_{ij} = 1, \quad i = 1, \ldots, N.$$

The difference between the two variables is λ_{ij} is continuous. Then, similar to the work in [18], we use a series of reweighted linear functions $\chi(\lambda_{ij}) = \frac{\lambda_{ij}}{\lambda_{ij}^{n-1}+\delta}$, where δ is a small positive constant and n is the number of iterations, to replace Ω in the (39). Then, we can find the optimization problem in (39) becomes a convex optimization problem after modification. The algorithm in [18] and [19] directly return the λ_{ij}^* as the finally result when it is convergence, however, this way of update can not guarantee the algorithm converges to the optimal solution, since the change of any user's decision will change the matrix $\boldsymbol{R}$ in turn. When a lot of users make their decision simultaneously with a fixed $\boldsymbol{R}$, this will certainty lead to obtain a non-optiaml solution and make oscillation. To be specific, when the matrix $\boldsymbol{R}$ is fixed, all the users will choose to offload through the subchannel which has smaller number of users, which leads to the subchannel that has smaller number of users originally will get more users, and the same situation will happen in the next iteration in turn. Based on the problem above, we make some modifications to the algorithm in order to obtain the optimal solution. Firstly, we update the uplink date rate matrix $\boldsymbol{R}$ in the solving process. Secondly, we do not update the decision matrix $\boldsymbol{\Omega}$ directly, but update a fixed number users randomly. This is similar to the way we use in scalar updating, since there is always a learning rate in the scalar solving process. These modifications make the algorithm more robust and have a higher convergence rate. The entire algorithm for offloading scheme problem is summarized in Algorithm 2.

It is worth mentioning that we only substitute the linear function $\chi(\lambda_{ij})$ for $\boldsymbol{\Omega}$ in the objective function (39a) and replace the constraint (39b) by $\lambda_{ij} \geq 0$, and replace the variable ω_{ij} by λ_{ij} in the constraint (39c).

Algorithm 2 Offloading Scheme Algorithm

```
Initialization:
Set $n = 0$, $\lambda_{ij}^0 = 1 - \delta$, where $\delta$ is a small positive constant.
Given a convergence threshold $\epsilon$.
end initialization

repeat
    Set $n = 0$, $\lambda_{ij}^0 = 1 - \delta$.
    Update $\boldsymbol{R}$ by (38).
    repeat Set $n = n + 1$
        Let $\left\{\lambda_{ij}^{n-1}\right\}$ denote the solution in the last iteration, and denote the linear function as $\chi(\lambda_{ij}) = \frac{\lambda_{ij}}{\lambda_{ij}^{n-1}+\delta}$, $i = 1, \ldots, N, j = 1, \ldots, N + M$.
        Replace the variable $\boldsymbol{\Omega}$ in the objective function (39a) by $\chi(\lambda_{ij})$, then solve the modified convex optimization problem to obtain $\left\{\lambda_{ij}^n\right\}$.
    until $\left|\lambda_{ij}^n - \lambda_{ij}^{n-1}\right| < \epsilon$, return $\left\{\lambda_{ij}^n\right\}$
    Let $\boldsymbol{d}^n$ denote the desicion vector at $n$-th iteration, which can be obatined through $\left\{\lambda_{ij}^n\right\}$.
    Generate a random 0-1 vector $\boldsymbol{\theta}$, which has a fixed number of ones.
    Update the decision vector according to
        $$\boldsymbol{d}^n = \boldsymbol{d}^{n-1} + \boldsymbol{\theta} \circ \boldsymbol{d}^n$$
until $\boldsymbol{d}^n$ approximately equals $\boldsymbol{d}^{n-1}$.
```

Since the problem (39) is a convex problem after modified, Algorithm 2 will converge to the globally optimal solution. Here, we explain why we can use the modified problem to approximate the the problem in (39). Note that $\left|\lambda_{ij}^n - \lambda_{ij}^{n-1}\right| < \epsilon, \lambda_{ij}^n \approx \lambda_{ij}^{n-1}$ when the inner loop converges, and δ is a small positive constant. So, we can get result as follows:

$$\chi(\lambda_{ij}^n) = \frac{\lambda_{ij}^n}{\lambda_{ij}^{n-1} + \delta} \approx \begin{cases} 1 & \textit{if } \lambda_{ij}^n > 0 \\ 0 & \textit{if } \lambda_{ij}^n = 0 \end{cases}$$

which approximately equals to $\boldsymbol{\Omega}$. Then, the outer loop ensure the algorithm converges to the globally optimal solution.

C. CONVERGENCE ANALYSIS

In the following, we provide a proof for the convergence of the alternative method for the original optimization problem. $f(\Psi, \boldsymbol{d}, \boldsymbol{p})$ is the original objective function. Given the decision tuple $(\Psi^0, \boldsymbol{d}^0) \in \mathbb{M}$, based on Algorithm 1, we can decompose $f(\Psi^0, \boldsymbol{d}^0, \boldsymbol{p})$ into a series of convex sub-problems with respect to $\boldsymbol{p}$. Therefore, $\exists \boldsymbol{p}^* \in \mathbb{P}$, and we can obtain the following inequality

$$f(\Psi^0, \boldsymbol{d}^0, \boldsymbol{p}^*) \leq f(\Psi^0, \boldsymbol{d}^0, \boldsymbol{p}) \tag{40}$$

Then, based on the solution above, let $\boldsymbol{p} = \boldsymbol{p}^* \in \mathbb{P}$. Based on Algorithm 2, $\exists(\Psi^*, \boldsymbol{d}^*) \in \mathbb{M}$, and the following inequality

TABLE 1. Simulation parameters.

Parameters	Value
Number of mobile users, N	30
Number of subchannels, M	5
channel bandwidth, B	5 MHz
Transmission Power, p	100mWatts
Background noise, σ^2	-100 dBm
Path loss factor, β	3
Data size for the offloading, s_n	5 MB
Number of CPU cycles for one task, c_n	1 Gigacycles
Coefficient for hardware architecture, κ	3.5×10^{-27}
Convergence threshold, ϵ	1×10^{-4}

holds

$$f(\Psi^*, \boldsymbol{d}^*, \boldsymbol{p}^*) \leq f(\Psi, \boldsymbol{d}, \boldsymbol{p}^*) \tag{41}$$

From (40) and (41), we can obtain the following inequality

$$f(\Psi^*, \boldsymbol{d}^*, \boldsymbol{p}^*) \leq f(\Psi, \boldsymbol{d}, \boldsymbol{p}^*) \leq f(\Psi, \boldsymbol{d}, \boldsymbol{p}) \tag{42}$$

Thus, we can obtian the optimal solution $f(\Psi^*, \boldsymbol{d}^*, \boldsymbol{p}^*)$ by solving the two sub-problems alternatively.

V. NUMERICAL RESULTS

In this section, we present various numerical results to illustrate the the performance of our proposed algorithm. We assume that there is a wireless small-cell that the base-station's converage range is 50m [35], and the mobile device users are randomly scattered in the converge range. Based on the wireless interference model for urban cellular radio environment [36], we model the channel gain as: $h_{n,s} = L_{n,s}^{-\beta}$, where β is the path loss factor. Accounting for the diversity of computational capacitor among different mobile device, f_n^L is assgined from the set $\{0.5, 0.8, 1.0\}$ GHz randomly, and the computational resource allocated from edge cloud for user n is $f_n^C = 10$ Ghz. In the consideration of decision weights for execution latency and energy consumption, we set $\alpha_n^t = 1 - \alpha_n^e$ and α_n^e is assigned from $\{0.1, 0.5, 0.9\}$ randomly. The other settings of simulation parameters are given in Table 1.

We compare our proposed offloading algorithm with three different strategies as follows:

- All Local Computing Scheme: all the users choose to compute its tasks locally. This scenario corresponds to the current situation that there is no available edge cloud, and we need to compute the tasks on our own mobile devices.
- All Edge Computing Scheme: all the users choose to offload the tasks to edge cloud for processing through a randomly selected channel, while uses the same transmission power control method as our proposed. This scenario corresponds to an extreme situation that all the users are irrational, and they all want to use the edge resources as much as possible.
- Fixed Transmission Power Scheme: the user offloads the task through a fixed transmission power, while uses the same offloading scheme as our proposed. We design this strategy to test the effect of our power control algorithm.

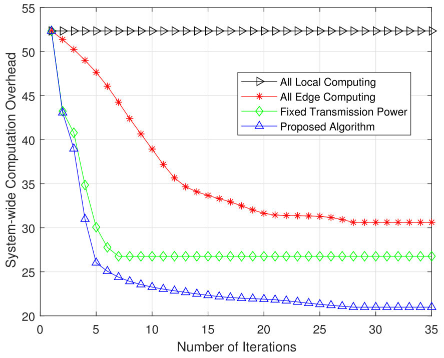

FIGURE 3. Simulation dynamics of system-wide computation overhead of different offloading scheme or power control method.

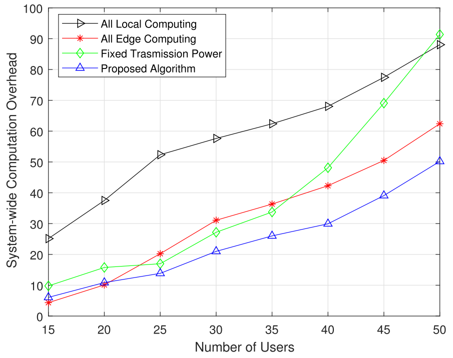

FIGURE 4. Average system-wide computation overhead with different number of users.

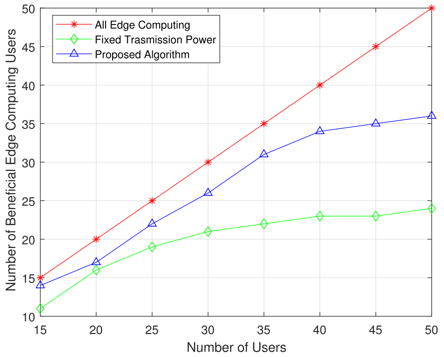

FIGURE 5. Average number of beneficial edge computing users with different number of users.

We first show the dynamics of the system-wide computation overhead of four different offloading scheme or power control method in Fig. 3. We can find that all the schemes can keep the computation overhead decreasing until convergence, and our proposed algorithm can reduce 60% and 32%, and 19% system-wide computation overhead compared with the schemes including all local computing and all edge computing and fixed transmission power, respectively.

In Fig. 4 and Fig. 5, we show average system-wide computation overhead and average number of beneficial edge computing users with different number of users $N = 15, 20, \ldots, 50$ [], respectively. We can find that our proposed

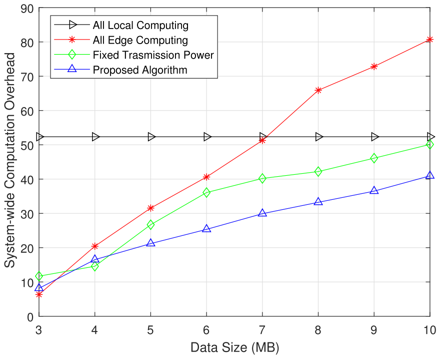

FIGURE 6. Average system-wide computation overhead with different data size.

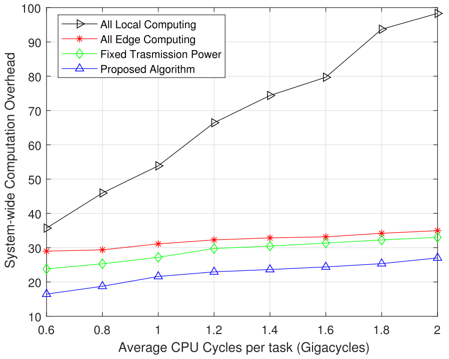

FIGURE 8. Average system-wide computation overhead with different CPU cycles.

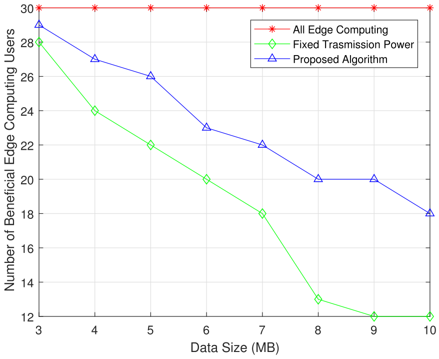

FIGURE 7. Average number of beneficial edge computing users with different data size.

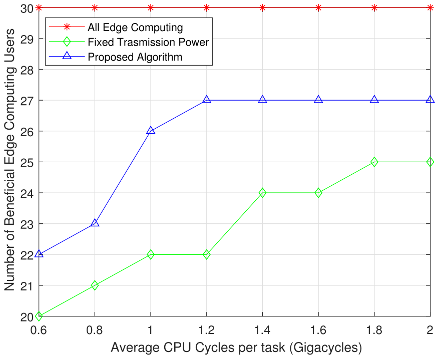

FIGURE 9. Average number of beneficial edge computing users with different CPU cycles.

algorithm can reduce 58% and 23%, and 36% system-wide computation overhead compared with the schemes including all local computing and all edge computing and fixed transmission power, respectively. As for the aspect of number of beneficial edge computing users, our proposed algorithm can achieve up-to 35% improvement over the scheme by fixed transmission power.

In Fig. 6 and Fig. 7, we show average system-wide computation overhead and average number of beneficial edge computing users with different data size $s = 3, 4 \ldots, 10$ (MB), respectively. We can find that our proposed algorithm can reduce 49% and 42, and 21% system-wide computation overhead compared with the schemes including all local computing and all edge computing and fixed transmission power, respectively. Since the data size is irrelevant to the user's overhead who computes locally, the system-wide computation overheads for all local computing with different date size are the same. As for the aspect of number of beneficial edge computing users, our proposed algorithm can achieve up-to 24% improvement over the scheme by fixed transmission power. As the data size grows, the users tend to compute locally.

In Fig. 8 and Fig. 9, we show average system-wide computation overhead and average number of beneficial edge computing users with different CPU cycles $c = 0.6, 0.8 \ldots, 2$ (Gigacycles), respectively. We can find that our proposed algorithm can reduce 76% and 32%, and 30% system-wide computation overhead compared with the schemes including all local computing and all edge computing and fixed transmission power, respectively. Since the increase of CPU cycles has comparatively small impact on the user's overhead who computes on edge cloud, the system-wide computation overheads for all schemes have a slow increase tendency. As for the aspect of number of beneficial edge computing users, our proposed algorithm can achieve 12% improvement over the scheme by fixed transmission power. As the CPU cycles grows, the users tend to compute on edge cloud.

Comparing with other schemes, we can conclude that our proposed algorithm can reduce the system-wide computation overhead or improve the number of users who can benefit from edge cloud efficiently. The main difference between our proposed method and [19] is that we do not update the decision matrix $\mathbf{\Omega}$ directly, but update a fixed number users randomly, which makes our method have a faster convergence speed. And [14] formulates the offloading decision making problem as a game, and only considers the offloading scheme problem.

VI. CONCLUSION

In this paper, we first formulate the problem as a mixed integer non-linear optimization problem which is difficult to solve it directly. Then, we utilize the alternative optimization method to slove this joint problem by decomposing it into two sub-problems, and we design two algorithms to slove these two sub-problems respectively. Numerical results show that our proposed scheme is more efficient compared with others schemes mentioned in this paper.

In future work, we will consider the users mobility, which is more challenging and general.

REFERENCES

[1] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2012, pp. 59–66.

[2] Z. Chen, W. Hu, J. Wang, S. Zhao, B. Amos, G. Wu, K. Ha, K. Elgazzar, P. Pillai, R. Klatzky, D. P. Siewiorek, and M. Siewiorek, "An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance," in *Proc. 2nd ACM/IEEE Symp. Edge Comput.*, Oct. 2017, pp. 1–14.

[3] D. Tian, J. Zhou, Z. Sheng, M. Chen, Q. Ni, and V. C. M. Leung, "Self-organized relay selection for cooperative transmission in vehicular ad-hoc networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9534–9549, Oct. 2017.

[4] C. Dai, X. Liu, J. Lai, P. Li, and H.-C. Chao, "Human behavior deep recognition architecture for smart city applications in the 5G environment," *IEEE Netw.*, to be published.

[5] Y. Wu, L. P. Qian, J. Zheng, H. Zhou, and X. S. Shen, "Green-oriented traffic offloading through dual connectivity in future heterogeneous small cell networks," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 140–147, May 2018.

[6] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 771–786, Apr. 2018.

[7] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[8] *Mobile-Edge Computing—Introductory Technical White Paper*, Eur. Telecommun. Standards Inst., Sophia Antipolis, France, Sep. 2014.

[9] U. Drolia, R. Martins, J. Tan, A. Chheda, M. Sanghavi, R. Gandhi, and P. Narasimhan, "The case for mobile edge-clouds," in *Proc. IEEE 10th Int. Conf. Ubiquitous Intell. Comput.*, Dec. 2013, pp. 209–215.

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

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

[12] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[13] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[14] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[15] K. Shen and W. Yu, "Fractional programming for communication systems—Part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, May 2018.

[16] J. J. Escudero Garzás, M. Hong, A. Garcia, and A. Garcia-Armada, "Interference pricing mechanism for downlink multicell coordinated beamforming," *IEEE Trans. Commun.*, vol. 62, no. 6, pp. 1871–1883, Jun. 2014.

[17] C. Shi, R. A. Berry, and M. L. Honig, "Distributed interference pricing with MISO channels," in *Proc. IEEE 46th Annu. Allerton Conf. Commun., Control, Comput.*, Sep. 2008, pp. 539–546.

[18] Y. Liu, D. Niu, and B. Li, "Delay-optimized video traffic routing in software-defined interdatacenter networks," *IEEE Trans. Multimedia*, vol. 18, no. 5, pp. 865–878, May 2016.

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

[20] M.-H. Chen, B. Liang, and M. Dong, "Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2017, pp. 1–9.

[21] X. Lyu, H. Tian, P. Zhang, and C. Sengul, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2016.

[22] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[23] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[24] 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.

[25] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 319–333, Feb. 2019.

[26] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst.*, vol. 13, nos. 2–3, pp. 203–221, Aug./Sep. 1996.

[27] 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.

[28] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. USENIX Conf. Usenix Tech. Conf.*, 2010, p. 21.

[29] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[30] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, Nov. 2017.

[31] K.-H. Loh, B. Golden, and E. Wasil, "Solving the maximum cardinality bin packing problem with a weight annealing-based algorithm," in *Operations Research and Cyber-Infrastructure*. Boston, MA, USA: Springer, 2009, pp. 147–164.

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

[33] A. Alvarado, G. Scutari, and J.-S. Pang, "A new decomposition method for multiuser DC-programming and its applications," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2984–2998, Jun. 2014.

[34] C. Pan, W. Xu, W. Zhang, J. Wang, H. Ren, and M. Chen, "Weighted sum energy efficiency maximization in ad hoc networks," *IEEE Wireless Commun. Lett.*, vol. 4, no. 3, pp. 233–236, Jun. 2015.

[35] T. Q. S. Quek, G. de la Roche, I. Güvenç, and M. Kountouris, *Small Cell Networks: Deployment, PHY Techniques, and Resource Management*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[36] T. S. Rappaport, *Wireless Communications: Principles and Practice*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

JUN LIU is currently a Researcher with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, China. His research interests include wireless communication, edge computing, and deep learning.

PAN LI is currently a Professor with the School of Communication and Information Engineering, University of Electronic Science and Technology of China, China. His research interests include big data networks, cloud computing, and artificial intelligence.

JIANQI LIU is currently pursuing the master's degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include big data analysis, deep learning, and edge computing.

JINFENG LAI is currently with the School of Information and Communication Engineering, University of Electronic Science and Technology of China. He has authored or coauthored over 100 refereed papers in journals, conferences, and workshop proceedings about his research areas within four years. His research interests include multimedia communications, sensor-based healthcare, and embedded systems. He is a member of the IEEE CIRCUITS AND SYSTEMS and the IEEE Communications Societies.