Received February 27, 2020, accepted March 8, 2020, date of publication March 11, 2020, date of current version March 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2980032

Joint Wireless Source Management and Task Offloading in Ultra-Dense Network

SHANCHEN PANG AND SHUYU WANG

College of Computer Science and Technology, China University of Petroleum, Qingdao 266580, China

Corresponding author: Shanchen Pang (pangsc@upc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672033, Grant 61873280, Grant 61873281, Grant 61972416, Grant 61672248, and Grant 61902430, in part by the National Key Research and Development under Project 2018YFC1406204, in part by the Key Research and Development Program of Shandong Province under Grant 2019GGX101067, in part by the Natural Science Foundation of Shandong Province under Grant ZR2019MF012, in part by the Taishan Scholars Fund under Grant ZX20190157, in part by the Independent Innovation Research under Project 18CX02152A, and in part by the Fundamental Research Funds for the Central Universities under Grant 19CX02028A.

ABSTRACT The ultra-dense network (UDN) based on mobile edge computing (MEC) is an important technology, which can achieve the low-latency of 5G communications and enhance the quality of user experience. However, how to improve the task offloading efficiency is a hot topic of UDN under the constraint on the limited wireless resources. In this article, we propose a heuristic task offloading algorithm HTOA to optimize the delay and energy consumption of offloading tasks in UDN. Firstly, a convex programming model for MEC resource allocation is established, which aims to obtain the optimal allocation set of resources for offloading tasks, and optimize the execution delay of offloading tasks. Followed by, the problem of joint channel allocation and user upload power control is solved by the greedy strategy and golden section method, which aims to optimization the delay and energy consumption of task upload data. Compared with the random task offloading algorithm, numerical simulations show that the algorithm HTOA can effectively reduce the delay and energy consumption of task offloading, and perform better as the number of users increases.

INDEX TERMS Ultra-dense network (UDN), mobile edge computing (MEC), task offloading.

I. INTRODUCTION

With the development of science and technology, the user equipment (UE) has connected to the wireless network are increasing explosively. According to the Cisco White Paper, monthly wireless data traffic will up to 77 EB/month in 2022, nearly 7 times higher than 2017, and the highest proportion of use will be UEs (about 90%) [1]. It can be seen that the use of UEs is main contribution for wireless traffic generation. Nowadays, the characteristics of UEs tend to be more intelligent and multi-media, which makes the emergence of new services and applications (such as augmented reality, wearable intelligent devices, Internet of vehicles, etc). However, such services and applications tasks cannot be completed only by the computation capability and residue battery's capacity of UE's. Furthermore, so as to complete tasks effectively, the research of task offloading strategy has become one of the hot topics. The initial task offloading is implemented in the mobile cloud computing (MCC) environments. Due to the deployment of MCC is far away from UEs, which cases that the delay and energy consumption highly for the tasks with offloading to MCC. Thereby, this still cannot meet the application requirements for the delay and energy consumption sensitive.

The concept of Mobile Edge Computing (MEC) was proposed by the European Telecommunications Standards Institute (ETSI) in 2014 [2]. The MEC is typically deployed on the base station around the UEs, and the distance between MEC and UEs are greatly shortened, which can provide the computation capability for UEs in a short time [3]. Therefore, the emergence of MEC provides a new idea to solve the application requirements for the delay and energy consumption sensitive [4]. In the white paper on 5G vision and requirements, the vision of achieving ''zero'' delay quality of experience (QoE) and low-power access for hundreds of millions of devices in 2020 and the future is described [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Shouguang Wang.

In order to achieve 5G vision, the main measures taken are to increase the system capacity, which include three ways, increasing spectrum bandwidth, improving spectrum utilization and cell splitting [6]. Among this, the cell splitting mode is recognized as the most effective way, which aims to supplement the blind areas that the macro base station (MBS) fails to cover by densely deploying low-power small base stations (SBS), and can improve the spatial reuse degree of cell. In particular, this model is more effective for hotspot areas, such as dense residential areas, train stations, stadiums, shopping malls and other crowded areas. In this context, the UDN has emerged [7].

It is an inevitable trend that the deployment of SBS with MEC servers under the architecture of UDN. The intensive deployment of SBS improving QoE of users for task execution and meeting the low-delay needs of future communication technology. Meanwhile, it also brings many problems and challenges, mainly including: (1) For MEC servers with limited computing resource, how to allocate computing resource reasonably to tasks with the character of pluralistic is a challenging problem. (2) The UEs tasks are offloading to MEC servers through different channels. When a large number of UEs chooses the same channel to compete for limited channel resource, which causes that the multiple tasks uplink transmission rates are reduced. Therefore, how to allocate the channels according to the channel spectrum resource status must be solved. (3) In the process of channel allocation, the delay and energy consumption of the UE task is affected by the corresponding upload power of task. The upload power control of the task effectively, it is a problem that should not be neglected. (4) Based on the radio resource information, it is a challenging problem for UEs to make decisions on whether to execute locally (their own devices) or MEC server (offloading). All in all, the UDN architecture has the characteristics of heterogeneous network complexity, the finiteness of the MEC server resources, the diversity of UEs task requirements, and the scarcity of spectrum resources. This characteristic causes that it has become a research hotspot, management of radio resources and task offloading strategy efficiently and effectively and with the delay and energy consumption as the research goal.

In this article, we research the problem of joint radio resources management and task offloading strategy in UDN. The main contributions are:

- We propose a heuristic task offloading algorithm HTOA, which updates radio resource management alternately in each iteration aims to maximize the sum of the users offloading benefits. The user offloading benefit is measured by the delay and energy consumption for task execution. The radio resource management includes the allocation of MEC computing resources, the channel allocation, and the user upload power control.
- The joint radio resources management and task offloading strategy problem is a mixed integer non-linear programming (MINLP). Therefore, we decomposed the original problem into two sub-problems by uses the idea of divide and conquer, so as to reduce the complexity of problem solving. The two sub-problems are the allocation problem of the MEC computing resource and joint the channel allocation and upload power control problem.
- We set up a convex programming model in view of the problem of allocation of MEC computing resources, which the KKT condition is used to get the optimal solution set of resource allocation for offloading tasks, and optimizes the execution delay of offloading tasks on MEC servers.
- We decompose the problem of channel allocation and upload power control into two sub problems: the problem of channel allocation and the problem of upload power control, which are solved by greedy strategy and golden section method respectively, and optimize the delay and energy consumption of task upload data.
- The proposed algorithm HTOA is evaluated through the large number of simulation experiments. The experimental results show that the algorithm HTOA is better than the random offloading algorithm in the sum of user offloading benefits. It's noted that, the advantage is more obvious after the number of users exceeds 70, which indicates that the algorithm HTOA is more suitable for the task offloading of multi-user in UDN. At the same time, by selecting different performance parameters, the impact on the users offloading benefit is evaluated.

The rest of this article is structured as follows. We first reviews related works in Section II. The system model is described in Section III. For the joint radio resource management and task offloading, we propose heuristic task offloading algorithm (HTOA) in Section IV. Our simulation parameters setting and discussions are given in Section V. Finally, Section VI concludes this article.

II. RELATED WORKS

For task offloading, delay, energy consumption or the trade-off between them is often used as performance standards to measure the computing overhead by many scholars in recent years. For the multi-user scenarios, Chen [8] proposed a decentralized offloading strategy based on the game theory (GA), which aimed to minimize the delay and energy consumption. The decentralized offloading strategy was further enriched in literature [9] based on this, and the concept of beneficial cloud computing users was proposed, which means that the users choose to offloading and the computing overhead after offloading is less than the local execution. For the single user scenarios, Liu *et al.* [10] analyzed the task execution delay based on Markov decision theory, which the dynamic task buffer queue was established for not executed tasks and then scheduled tasks from the queue based on relevant state information, finally obtain the minimizing delays. The above researches are applicable to the scenarios of the single MEC. From the realistic perspective, the scenarios of UDN combined with the multiple MEC servers are more popular. Guo *et al.* [11] proposed the offloading strategy of the optimal enumeration and the two-level game offloading strategy combining with greedy ideas, so as to optimization the weighted sum of delay and energy consumption. Yang *et al.* [12] proposed the task offloading algorithm based on the idea of games in UDN, which mainly aimed at saving energy consumption with the constraint of delay. From the perspective of improving the computing overhead of green MEC, Chen *et al.* [13] designed an MEC server with an energy harvesting to serve the mobile user's task offloading needs, and proposed centralized and decentralized offloading algorithms respectively. Bottai *et al.* [14] investigated the problem of energy consumption for user terminals, which a long-term evolution UDN model has been generated and an energy consumption correlation algorithm was designed. For the multi-task scenarios with sequential constraints, sun *et al.* [15] proposed an energy-aware mobility management algorithm, and the cost of switching between tasks is considered. In addition, the literatures [16]–[23] also investigated the problem of task offloading, but the environment is cloud system.

The researches of joint radio resources management and task offloading strategy has more widely concerned by scholars. Mao *et al.* [24] proposed a task offloading strategy based on Lyapunov optimization, which the user upload power and CPU cycle frequency are taken into account. Furthermore, energy consumption optimization problem of the mobile device and MEC server-side was investigated in the literature [25], and the constrained of task delay was considered. Lyu *et al.* [26] proposed the semi-distributed offloading strategy to jointly optimization the resource management and task offloading decisions, which can effectively reduce the computational utility of users and systems. Similarly, Ren *et al.* [27] investigated the problem of resource allocation and task offloading, which aims to minimize the end-to-end delay. You *et al.* [28] take minimizing energy consumption as the performance standard, and studied the resource allocation and offloading decisions. Mao *et al.* [29] proposed sub-optimal algorithm based on the flow shop scheduling theory, which the convex optimization theory was used to optimize computing resources. Considering the impact of task dynamic arrival on task queuing delay, Chen *et al.* [30] proposed the offloading algorithm TOFFEE to minimizing energy consumption, which allocated the local computing resources and MEC computing resources respectively. The above researches are focusing on joint the radio resource management and task offloading strategy in the scenarios of single MEC server, which cause that the UDN emerged as a major technology in the 5G communication networks. In the multi-cell MEC application scenarios, Tran and Pompili [31] proposed the task offloading algorithm, and the user upload power allocation and the computing resource allocation are considered. Zhang *et al.* [32] proposed a task offloading algorithm, which the local computing resources, the channel resources, and user upload power resources are considered. Chen and Hao [33] constructed an UDN architecture that combines with software-defined networking (SDN), and then proposed the centralized task offloading algorithm to solve the task delay minimization problem. Guo *et al.* [34] allocated the local computing capacity of the mobile device for multi tasks, and optimized the upload power and task offloading strategy of the mobile devices, which the computational overhead of the devices is reduced effectively. Dai *et al.* [35] proposed the J-UACO algorithm so as to optimize the energy consumption, which the allocation of computing resources and upload power was considered. Li *et al.* [36] implemented an energy efficiency aware offloading mechanism that takes the channel allocation into consideration. Ning *et al.* [37] designed the offloading algorithm with delay as the optimization objective, which it is joint optimization the channel allocation, the upload power, and the task offloading.

To the best of our knowledge, few literatures considered the difference of measurement units between delay (unit second, S) and energy consumption (unit joule, J) [26], [31]. The application scenario in literature [26] has only one MEC server, which cannot meet the need of 5G communication. The application scenario in literature [31] is the UDN, but the management of its spectrum resources has not been considered. In this paper, we use the weighted sum of delay and energy consumption as the measurement standard, and take into account the difference of measurement units between delay (unit second, S) and energy consumption (unit joule, J). In addition, we joint considers the management of user upload power, the MEC server computing resources, and the channel allocation. This can reduce the delay and energy consumption of the UEs task for execution effectively.

III. SYSTEM MODEL

We consider the architecture of the UDN combined with the multiple MEC servers, and the specific architecture is shown in Figure 1. We assume that the set of SBS (represent MECs) is $\mathcal{S} = \{1,2,\ldots,S\}$, the set of UEs is $\mathcal{N} = \{1,2,\ldots,N\}$, and each UE has a computing task. And each of the task is atomic and cannot be divided into subtasks with similar to literature [11], [26] and [38]. We use the method of Orthogonal Frequency Division Multiple Access (OFDMA) as the multiple access method for uses [39]. Divide the bandwidth into C sub-channels, and the bandwidth of channel can be described to $W = B/C$. In addition, we assume that UEs do not move during the execution of the task. The system model is established from three aspects: local computing, MEC server computing and multi-user offloading benefits. For ease of understanding, Table 1 shows the notations and their related descriptions used in this article.

A. THE MODEL OF LOCAL CALCULATION

We use two tuples $T_i = (w_i, s_i)$ to represent the computing task of each UE, where w_i represents the required computing capacity of task T_i (quantified by CPU cycles frequency), and s_i represents the data size of the computing task T_i (including program code, input file and other information). The task has only two state variables $\lambda_{i,j} \in \{0, 1\}, \forall i \in \mathcal{N}, j \in \mathcal{S}$.

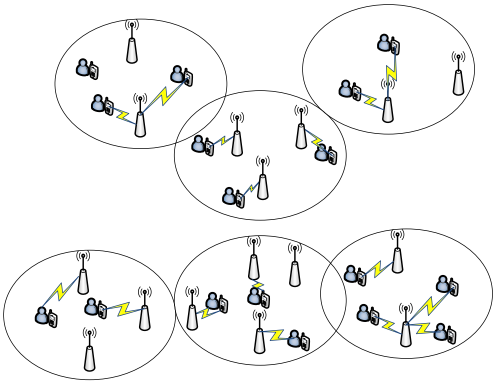


FIGURE 1. Architecture of UDN combined with multiple MEC servers.

TABLE 1. Units for magnetic properties.

Notation	Description
T_i	The task of the ith UE,$\forall i \in \mathcal{N}$
w_i	The required computation capability of task T_i
s_i	The data size of task T_i
$\lambda_{i,j}$	The status variable of task T_i, $\forall i \in \mathcal{N}, j \in \mathcal{S}$
f_i^l	The local computation capability of UE i
μ_i	The power coefficient of the task T_i energy consumption per CPU cycle
t_i^l	The delay of task T_i executing locally
e_i^l	The energy consumed of task T_i executing locally
p_i	The upload power of task T_i
$g_{i,j}$	The channel gain of offloading from task T_i to MEC j
$SNR_{i,k}$	The SNR of task T_i via wireless channel k
σ^2	The Gaussian noise of channel
$r_{i,k}$	The offloads rate of task T_i to MEC j
$t_{i,k}^{off}$	The delay of task T_i uploading via wireless channel k
$e_{i,j}^{off}$	The energy consumption of Task T_i offloading
$x_{i,j}^c$	The proportion of computation capability allocated by MEC j to task T_i
f_j^c	The computing capacity of MEC j
$t_{i,j}^{exe}$	The execute delay of task T_i execution on MEC j
$e_{i,j}^c$	The total energy consumption of offloading task T_i
$t_{i,j}^c$	The total delay of offloading task T_i
λ_i^t	The weight parameter of task T_i for delay demand
λ_i^e	The weight parameter of task T_i for energy consumption demand
$p_i^{\max}$	The maximum power of task T_i

When $\lambda_{i,j} = 1$, which indicates that the task T_i will offloading to MEC server through channel for complete execution. Otherwise, the task executing locally.

When $\lambda_{i,j} = 0$, which means that task T_i is executed locally by the UE. Computing resource of UE defined as f_i^l (quantified by CPU cycles frequency). Hence, the local computation delay of task T_i can be expressed as:

$$t_i^l = \frac{w_i}{f_i^l}. \tag{1}$$

While executing the task, it also needs to consume the remaining battery energy of the UE itself. The local energy consumption of task $T_i$can be defined as:

$$e_i^l = \mu_i w_i, \tag{2}$$

where μ_i represents power coefficient of the energy consumed by each CPU cycle. We use the models applied $u_i = \gamma \left(f_i^l\right)^2$ in the literature [8] and [40], and the size of γ depends on the chip structure of the UEs.

B. THE MODEL OF MEC CALCULATION

When $\lambda_{i,j} = 1$, which means that task T_i is offloaded to MEC j for execution. At this time, it is mainly divided into two phases: the first phase is the task data upload phase, and the second phase is the task execution phase. Next, the model establishment of MEC is explained in detail from two stages.

1) THE PHASE OF TASK DATA UPLOAD

In this phase, when the task T_i is offloaded through the wireless channel k, let $\gamma_{i,k} = 1$. The signal-to-noise ratio (SNR) $SNR_{i,k}$, which is specifically defined as:

$$SNR_{i,k} = \frac{p_i g_{i,j}}{\sigma^2 + \sum\limits_{n \in \mathcal{N}/\{i\}} \sum\limits_{s \in \mathcal{S}/\{j\}} \gamma_{n,k} p_n g_{n,s}}. \tag{3}$$

The p_i is the upload power of task T_i, it can be determined by the MEC similar to [29], [31], and $g_{i,j}$ is the channel gain offloading by task T_i to MEC j. The σ^2 represents the Gaussian noise of channel, $\sum\limits_{n \in \mathcal{N}/\{i\}} \sum\limits_{s \in \mathcal{S}/\{j\}} \gamma_{n,k} p_n g_{n,s}$ represents the inter-cell interference generated by other UEs in channel k.

We calculate the task offloading rate according to the Shannon equation after determining the $SNR_{i,k}$, which is specifically defined as:

$$r_{i,k} = W \log_2(1 + SNR_{i,k}). \tag{4}$$

Moreover, the offloading delay $t_{i,k}^{off}$ of task T_i can be expressed as:

$$t_{i,k}^{off} = \frac{s_i}{\gamma_{i,k} r_{i,k}}, \tag{5}$$

where s_i represents the data size of task T_i.

The UE needs to consume energy $e_{i,k}^{off}$.while offloading tasks, which it is can be defined as:

$$e_{i,k}^{off} = p_i t_{i,k}^{off}, \tag{6}$$

where p_i represents the upload power of task T_i.

2) THE PHASE OF TASK EXECUTION

The execution delay $t_{i,j}^{exe}$ of task T_i on MEC j can be defined as:

$$t_{i,j}^{exe} = \frac{w_i}{x_{i,j}^c f_j^c}, \tag{7}$$

where $x_{i,j}^c$ ($x_{i,j}^c \leq 1$) is the proportion of computing resource allocated to task T_i by MEC j, and f_j^c is the computing resource of MEC.

We ignore the backhaul delay and energy consumption of task execution result, which the reason is that the task execution result data is so small. In addition, the user-centered research in this paper does not consider the energy consumption of the system during the execution of the task in MEC.

Thus, the delay $t_{i,j}^c$ and energy consumption $e_{i,j}^c$ of the task offloading to MEC j are:

$$t_{i,j}^c = t_{i,k}^{off} + t_{i,j}^{exe}, \tag{8}$$

$$e_{i,j}^c = e_{i,k}^{off}. \tag{9}$$

C. OPTIMIZATION MODEL OF MULTI-USER OFFLOADING BENEFITS

In this sub-section, we propose optimization model of users offloading benefits. We definition the w_i^t and w_i^e represents the weight parameters of the UE's task delay and energy consumption respectively, which the sum of these is 1 ($w_i^t + w_i^e = 1$). When the UE has a high requirement for delay, increase the value of w_i^t, and correspondingly decrease the value of w_i^e, otherwise, vice versa. According to the model of local calculation and the MEC calculation, the user offloading benefits can be formulated as:

$$\mathrm{P} \quad \max f = \sum_{i=1}^{N}\sum_{j=1}^{S}\left(w_i^t \frac{t_i^l - \lambda_{i,j} t_{i,j}^c}{t_i^l} + w_i^e \frac{e_i^l - \lambda_{i,j} e_{i,j}^c}{e_i^l}\right) \tag{10}$$

$$s.t.\ \gamma_{i,k} \in \{0,1\}, \quad \forall i \in \mathcal{N},\ k \in \mathcal{C} \tag{11}$$

$$\sum_{k=1}^{C} \gamma_{i,k} \le 1, \quad \forall i \in \mathcal{N} \tag{12}$$

$$\lambda_{i,j} \in \{0,1\}, \quad \forall i \in \mathcal{N},\ j \in \mathcal{S} \tag{13}$$

$$\sum_{j=1}^{S} \lambda_{i,j} \le 1, \quad \forall i \in \mathcal{N} \tag{14}$$

$$\sum_{i=1}^{n} \lambda_{i,j} x_{i,j}^c \le 1, \quad \forall j \in \mathcal{S} \tag{15}$$

$$0 \le p_i \le p_i^{\max}, \quad \forall i \in \mathcal{N}^{off} \tag{16}$$

$$\lambda_{i,j}\left(\sum_{k=1}^{C} \frac{s_i}{r_{i,k}} + \frac{w_i}{x_{i,j}^c f_j^c}\right) + (1-\lambda_{i,j})\frac{w_i}{f_i^l} \le T_i^{\max}, \quad \forall i \in \mathcal{N},\ j \in \mathcal{S} \tag{17}$$

The objective function $f(\mathbf{X}^c, \mathbf{P}, \gamma)$ is to maximize the weighted sum of the delay and energy consumption of all UE tasks. The first constraint (11) and the second constraint (12) indicate that the UE task uploads or does not upload and can only upload through one channel. The third constraint (13) and the fourth constraint (14) indicate that the UE can only execution tasks locally or offloaded to one MEC server. The fifth constraint (15) indicates that the proportion of computing resource $x_{i,j}^c$ allocated to offloaded tasks does not exceed the MEC total computing resource. The sixth constraint (16) indicates that the upload power p_i of the offloaded user does not exceed the value of the maximum power $p_i^{\max}$ of the UE. The seventh constraint (17) indicates that the total execution delay of the UE does not exceed the value of maximum delay $T_i^{\max}$.

IV. JOINT RADIO RESOURCE MANAGEMENT AND TASK OFFLOADING

We propose a heuristic task offloading algorithm (HTOA) based on the optimization model mentioned in the system model. We rewrite the multi-user offloading benefits function (10), which is described as:

$$\mathrm{P} \quad \max f = \sum_{i=1}^{N}\sum_{j=1}^{S}\sum_{k=1}^{C}\left(1 - \frac{\lambda_{i,j} w_i^t f_i^l}{x_{i,j}^c f_j^c} - \gamma_{i,k}\frac{\alpha_i + \beta_i p_i}{r_{i,k}}\right) \tag{18}$$

$$s.t.\ (11) \sim (15) \tag{19}$$

Among this, $\alpha_i = w_i^t \frac{s_i f_i^l}{w_i}$, $\beta_i = w_i^e \frac{s_i}{\mu_i w_i}$. The optimization problem P is a MINLP problem [29], [34]. It's worth noting that the computing resources allocation of the MEC server is irrelevant to the channel allocation and user upload power, but there is a direct relationship between the channel allocation and upload power control. Therefore, in order to reduce the complexity of problem solving, the above problems can be decomposed into two sub-problems by uses the idea of divide and conquer. The two sub-problems are the MEC computing resource optimization allocation problem and the channel allocation and upload power control problem.

A. THE OPTIMIZATION ALLOCATION OF MEC COMPUTING RESOURCE

We investigated the optimal allocation of MEC server computing resource, so as to optimize the execution delay of tasks with offloaded to MEC server. When the task T_i is offloaded to the MEC server for execution, which means that $\sum_{j=1}^{S} \lambda_{i,j} = 1$. We denote N_j ($N_j \le N$) as the number of tasks offloaded to the MEC j, and the set of offloading tasks is defined as $\mathcal{N}^{off} = \{1, 2, \ldots, N^{off}\}$. Given the vector γ of task offloading, transforming the original optimization problem P into the optimization allocation problem P_1 of MEC server computing resource:

$$\mathrm{P}_1 \quad \min_{\mathbf{X}^c} f(\mathbf{X}^c) = \sum_{j=1}^{S}\sum_{i=1}^{N_j} w_i^t \frac{f_i^l}{x_{i,j}^c f_j^c} = \sum_{j=1}^{S}\sum_{i=1}^{N_j} w_i^t \frac{f_i^l}{f_j^c}\frac{1}{x_{i,j}^c} \tag{20}$$

$$s.t.\ 0 x_{i,j}^c \le 1, \quad \forall i \in \mathcal{N}_j,\ j \in \mathcal{S} \tag{21}$$

The condition constraint (21) indicates that the proportion of computing resource $x_{i,j}^c$ allocated to offloaded tasks T_i does not exceed the MEC j total computing resource.

Lemma 1: Function $f(\mathbf{X}^c)$ is a convex programming problem for computing resource allocation matrix $\mathbf{X}^c$.

Proof: From problem P_1, we can see that $f(\mathbf{X}^c)$ is a multivariate function of the independent variable $\mathbf{X}^c$. The $\mathbf{X}^c$ is a matrix of $N^{off} \times S$, the row represents the number of offloading tasks, the column represents the number of MEC servers, and each item x_{ij}^c in the matrix $\mathbf{X}^c$ represents the proportion of computing resource allocated to the task T_i by the MEC j.

$$\mathbf{X}^c = \begin{bmatrix} x_{11}^c & x_{12}^c & \cdots & x_{1S}^c \\ x_{21}^c & x_{22}^c & \cdots & x_{2S}^c \\ \vdots & \vdots & & \vdots \\ x_{N^{off}1}^c & x_{N^{off}2}^c & \cdots & x_{N^{off}S}^c \end{bmatrix} \tag{22}$$

The proportion set $\mathbf{X}_j^c = \left(x_{1j}^c, x_{2j}^c, \ldots, x_{N_j j}^c\right)^{\mathrm{T}}, \forall j \in S$ of computing resource allocated by any MEC j, the second derivatives of $f(\mathbf{X}_j^c) = \sum_{i=1}^{N_j} w_i^t \frac{f_i^l}{f_j^c} \frac{1}{x_{i,j}^c}$ always exists for each independent variable, so we can get the following Hesse matrix $\mathbf{H}_j$ corresponding to any one MEC j.

$$\mathbf{H}_j = \begin{bmatrix} \frac{\partial^2 f}{\partial^2 x_{1j}^c} & \frac{\partial^2 f}{\partial x_{1j}^c \partial x_{2j}^c} & \cdots & \frac{\partial^2 f}{\partial x_{1j}^c \partial x_{N_j j}^c} \\ \frac{\partial^2 f}{\partial x_{2j}^c \partial x_{1j}^c} & \frac{\partial^2 f}{\partial^2 x_{2j}^c} & \cdots & \frac{\partial^2 f}{\partial x_{2j}^c \partial x_{N_j j}^c} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_{N_j j}^c \partial x_{1j}^c} & \frac{\partial^2 f}{\partial x_{N_j j}^c \partial x_{2j}^c} & \cdots & \frac{\partial^2 f}{\partial^2 x_{N_j j}^c} \end{bmatrix} \tag{23}$$

Any one of the second derivatives is:

$$\frac{\partial^2 f}{\partial x_{ij}^c \partial x_{nj}^c} = \begin{cases} 2w_i^t \frac{f_i^l}{f_j^c (x_{ij}^c)^3} & \mathit{if}\ i = n \\ 0 & \mathit{otherwise} \end{cases} \tag{24}$$

From the above second derivatives, we can see that $\frac{\partial^2 f}{\partial x_{ij}^c \partial x_{nj}^c} \geq 0$ and all elements are 0 except the main diagonal. It shows that every element of the Hesse matrix diagonal is greater than 0 (i.e., every eigenvalue of the Hesse matrix is greater than 0). So the Hesse matrix is a symmetric positive definite matrix. According to the theory in literature [41], the Hesse matrix $\mathbf{H}_j$ is a positive definite matrix for each feasible solution, which means that the function $f(\mathbf{X}_j^c)$ is a strictly convex function. It can be concluded that the computing resource allocation problem $f(\mathbf{X}^c)$ is a convex programming problem. □

In addition, due to the function $f(\mathbf{X}^c)$ is strictly convex, which means that there is a single global minimum for the independent variable $\mathbf{X}^c = (\mathbf{X}_1^c, \mathbf{X}_2^c, \ldots, \mathbf{X}_S^c)$ (i.e., there is a unique optimal allocation value for the independent variable of the computing resource allocation).

Thereby, we can obtain the optimal solution by constructing a generalized Lagrangian function to solving a series of conditions that satisfy the KKT condition and equality constraints [42].

Theorem 1 Obtain the unique optimal allocation solution $x_{i,j}^* = \frac{\sqrt{w_i^t f_i^l}}{\sum_{i=1}^{N_j} \sqrt{w_i^t f_i^l}}$ for the independent variable $\mathbf{X}^c = (\mathbf{X}_1^c, \mathbf{X}_2^c, \ldots, \mathbf{X}_S^c)$ of computing resource allocation.

According to the Lemma1, the Lagrangian function of the optimal problem P_1 can be defined as:

$$L(\mathbf{x}_j^c, \delta) = \sum_{i=1}^{N_j} w_i^t \frac{f_i^l}{f_j^c} \frac{1}{x_{i,j}^c} - \delta(1 - \sum_{i=1}^{N_j} x_{i,j}^c). \tag{25}$$

Because,

$$\frac{\delta L(\mathbf{x}_j^c, \gamma)}{\delta x_{i,j}^c} = -\frac{w_i^t w_i}{\left(x_{i,j}^c\right)^2 f_i^c} + \delta, \tag{26}$$

so the KKT conditions and constraints of the optimized problem P_1 are:

$$\begin{cases} -w_i^t \frac{f_i^l}{f_j^c \left(x_{i,j}^c\right)^2} + \delta = 0 \\ 1 - \sum_{i=1}^{N_j} x_{i,j}^c = 0 \end{cases} \tag{27}$$

The above formula (24) is the KKT condition of P_1, where the second in formula (27) is called the complementary relaxation condition in the KKT condition for which there exists an optimal solution for computing the resource allocation matrix.

We can get the optimal $x_{i,j}^*$ as:

$$x_{i,j}^* = \frac{\sqrt{w_i^t f_i^l}}{\sum_{i=1}^{N_j} \sqrt{w_i^t f_i^l}}. \tag{28}$$

In the end, we get the optimal allocation set of $f(\mathbf{X}^*)$ for the computing resource allocation matrix $\mathbf{X}^c$ as follows:

$$\begin{aligned} f(\mathbf{X}^*) &= \sum_{j=1}^{S} \sum_{i=1}^{N_j} w_i^t \frac{f_i^l}{f_j^c} \frac{\sum_{i=1}^{N_j} \sqrt{w_i^t f_i^l}}{\sqrt{w_i^t f_i^l}} \\ &= \sum_{j=1}^{S} \frac{\left(\sum_{i=1}^{N_j} \sqrt{w_i^t f_i^l}\right)^2}{f_j^c \sqrt{w_i^t f_i^l}} \end{aligned} \tag{29}$$

□

B. THE CHANNEL ALLOCATION AND UPLOAD POWER CONTROL

In order to optimize the delay and energy consumption of task upload data, we rewrite the original problem P into the problem P_2 of channel allocation and upload power control under the condition of given the matrix $\mathbf{X}$ as follows:

$$\begin{aligned} \text{P}_2 \quad \min_{\mathbf{P},\gamma} f_2(\mathbf{P}, \gamma) &= \sum_{i=1}^{N} \sum_{k=1}^{C} \gamma_{i,k} \frac{\alpha_i + \beta_i p_i}{r_{i,k}} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{C} \gamma_{i,k} \frac{\alpha_i + \beta_i p_i}{W \log_2(1 + SNR_{i,k})} \end{aligned} \tag{30}$$

The problem P_2 is still the mixed integer non-linear problem, so we re-decompose the problem P_2 as the channel allocation sub-problem P_{21} and the upload power control sub-problem P_{22}, respectively.

1) THE CHANNEL ALLOCATION

Assuming that the upload power control matrix $\mathbf{P}$ is given, the channel allocation sub-problem P_{21} can be describe as:

$$\text{P}_{21} \quad \min_{\gamma} f_{21}(\gamma) = \sum_{i=1}^{N} \sum_{k=1}^{C} \gamma_{i,k} \left(\frac{\alpha_i + \beta_i p_i}{W \log_2(1 + SNR_{i,k})}\right)$$

$$=\sum_{i=1}^{N}\sum_{j=1}^{S}\sum_{k=1}^{C}\frac{\alpha_i+\beta_i p_i}{W}\frac{\gamma_{i,k}}{\log_2(1+p_i EI_i)} \tag{31}$$

$$s.t.\ \gamma_{i,k}\in\{0,1\},\quad \forall i\in\mathcal{N},\ k\in\mathcal{C} \tag{32}$$

$$\sum_{k=1}^{C}\gamma_{i,k}\le 1,\quad \forall i\in\mathcal{N} \tag{33}$$

$$\lambda_{i,j}\left(\sum_{k=1}^{C}\frac{s_i}{r_{i,k}}+\frac{w_i}{x_{i,j}^c f_j^c}\right)+(1-\lambda_{i,j})\frac{w_i}{f_i^l}\le T_i^{\max},\quad \forall i\in\mathcal{N},\ j\in\mathcal{S} \tag{34}$$

Amon this, $EI_{i,k}$ represents the effective interference for task T_i through wireless channel k, it can be describe as:

$$EI_{i,k}=\frac{g_{i,j}}{\sigma^2+\sum\limits_{n\in\mathcal{N}/\{i\}}\sum\limits_{s\in\mathcal{S}/\{j\}}\gamma_{n,k}p_n g_{n,s}}. \tag{35}$$

We can see that the channel allocation for the problem P_{21} is determined by the amount of effective interference $EI_{i,k}$ under the condition of given the upload power. The larger the effective interference, the greater the UE task T_i offloading benefit through the channel k.

The problem of channel allocation is to investigate the tasks in the offloading task set $\mathcal{N}^{off}$. At this time, the constraint (29) is further transformed as:

$$\frac{s_i}{r_{i,k}}+\frac{w_i}{x_{i,j}^c f_j^c}\le T_i^{\max}$$

$$\Leftrightarrow EI_{i,k}\ge\frac{1}{p_i}\left(2^{\frac{1}{Ws_i\xi_i}}-1\right),\quad \forall i\in\mathcal{N}^{off}, \tag{36}$$

among $\xi_i=T_i^{\max}-\frac{w_i}{x_{i,j}^c f_j^c}$.

We use argmax function based on greedy strategy to solve the channel allocation sub-problem P_{21}. Resulting, the channel allocation can be described as:

$$\gamma_{i,k}=1\Bigg|_{k=\arg\max EI_{i,k}\cap EI_{i,k}\ge\frac{1}{p_i}\left(2^{\frac{1}{Ws_i\xi_i}}-1\right)}. \tag{37}$$

2) THE UPLOAD POWER CONTROL

Assuming that the channel allocation matrix γ is given, the upload power control sub-problem P_{22} can be describe as:

$$P_{22}\ \min_{\mathbf{P}} f_{22}(\mathbf{P})=\sum_{i=1}^{N^{off}}\sum_{k=1}^{C}\frac{\alpha_i+\beta_i p_i}{W\log_2(1+EI_{i,k}p_i)} \tag{38}$$

$$0<p_i\le p_i^{\max},\quad \forall i\in\mathcal{N}^{off} \tag{39}$$

$$\lambda_{i,j}\left(\sum_{k=1}^{C}\frac{s_i}{r_{i,k}}+\frac{w_i}{x_{i,j}^c f_j^c}\right)+(1-\lambda_{i,j})\frac{w_i}{f_i^l}\le T_i^{\max},\quad \forall i\in\mathcal{N},\ j\in\mathcal{S} \tag{40}$$

Similar to the channel allocation sub-problem, the constraint (40) is further transformed as:

$$\frac{s_i}{r_{i,k}}+\frac{w_i}{x_{i,j}^c f_j^c}\le T_i^{\max}$$

$$\Leftrightarrow p_i\ge\frac{2^{\frac{s_i}{W\xi_i}}-1}{EI_{i,k}},\quad \forall i\in\mathcal{N}^{off},\ j\in\mathcal{S},\ k\in\mathcal{C}. \tag{41}$$

From the constraints (39) and (41), we can conclude that the upload power control p_i range of user is $p_i\in\left[\frac{2^{\frac{1}{ws_i\xi_i}}-1}{EI_{i,k}},p_i^{\max}\right],\forall i\in\mathcal{N}^{off}$. And according to the user power control problem P_{22}, it can be known that there is no significant influence on the power between users. Thus, we define the function $h(p_i)$ of the user i as:

$$h(p_i)=\frac{\alpha_i+\beta_i p_i}{W\log_2(1+EI_{i,k}p_i)} \tag{42}$$

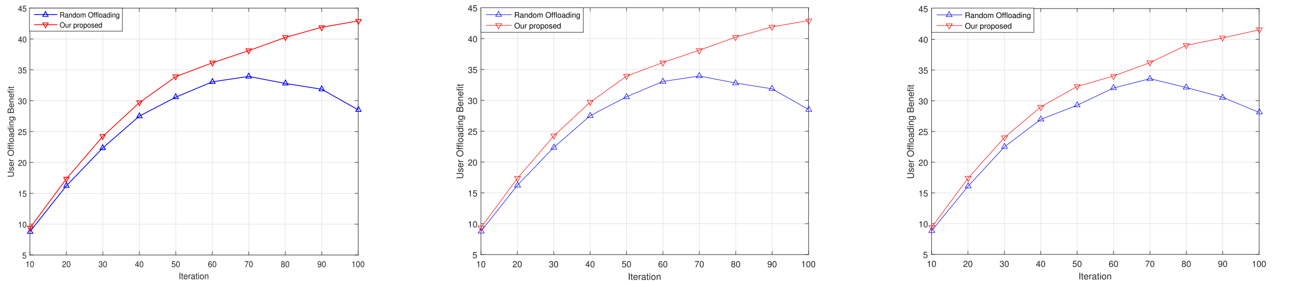
$$s.t.\ \frac{2^{\frac{1}{ws_i\xi_i}}-1}{EI_{i,k}}\le p_i\le p_i^{\max},\quad \forall i\in\mathcal{N}^{off} \tag{43}$$

The golden section method is used to calculate the user's upload power control. The pseudo code is shown in Algorithm 1. The input values include the search range $[a_i,b_i]$ of the user's uploaded power, the golden section point τ, and the convergence accuracy ε. The detailed steps can be described as follows: we firstly calculate the absolute value of the difference between the golden means p_i^l and p_i^r. When the absolute value is greater than the convergence accuracy, we calculate $h(p_i^1), h(p_i^2)$ by using (42) and compare the two values. If $h(p_i^1)>h(p_i^2)$, decrease research space to the left, otherwise decrease the search space to the right. The search space is continuously decreased to find the user optimal upload power by this method.

Algorithm 1 The Upload Power Control by Golden Section Method

Require: $a_i=p_i^{\min}=\frac{2^{\frac{1}{ws_i\xi_i}}-1}{EI_{i,k}}$, $b_i=p_i^{\max}$, $\tau=0.618$, $\varepsilon=0.05$
Ensure: p_i^{opt}

```
Calculate golden means p_i^l = b_i − τ*(b_i − a_i), p_i^r = a_i + τ*(b_i − a_i)
while |p_i^r − p_i^l| > ε do
    calculate h(p_i^1), h(p_i^2) by using (42)
    if h(p_i^1) > h(p_i^2) then
        a_i = p_i^l, p_i^l = p_i^r
        h(p_i^1) = h(p_i^2), p_i^r = a_i + τ*(b_i − a_i)
    else
        b_i = p_i^r, p_i^r = p_i^l
        h(p_i^2) = h(p_i^1), p_i^l = b_i − τ*(b_i − a_i)
    end if
end while
p_i^opt = (p_i^l + p_i^r)/2
```



(a) $s_i = 1000$ (b) $s_i = 2000$ (c) $s_i = 3000$

FIGURE 2. **The impact of task data size for user offloading benefits.**

Algorithm 2 The Heuristic Algorithm

Require: $\mathbf{P} = \mathbf{P}^{\max}$, $\lambda_{i,j} = 1, \forall i \in \mathcal{N}, j \in \mathcal{S}, \gamma_{i,k} = 0, \forall i \in \mathcal{N}, k \in \mathcal{C}$

calculated Local delay and energy consume by (1) and (2)
while iter <= itermax **do**
    calculated the proportion of computing resources matrix $\mathbf{X}^c$ by (29)
    channel selection matrix γ by (37)
    obtain the upload power control matrix $\mathbf{P}$ by using the algoritnm1
    update the effective interference by (35)
    calculated user offloading benefit $f(\mathbf{X}^c, \mathbf{P}, \gamma)$
    iter++
end while

C. THE HEURISTIC TASK OFFLOADING ALGORITHM

The specific flow of proposed heuristic task offloading algorithm HTOA is shown in Algorithm 2. Firstly, set each of task is offloading to MEC server randomly for execution, and we set the user upload power to the corresponding maximum of upload power. Meanwhile, we set no channel assigned to any tasks. Given the initialization, the delay and energy consumption of the local execution is calculated, and then the iteration of algorithm by the status of the wireless resources is started. In the process of iterative optimization, the computing resources of MEC servers are allocated to the tasks according to the formula (25) in the first step, which means that the tasks are executed by the MEC servers. The effective interference for channel allocation based on the greedy strategy is gained in the second step. The Algorithm 1 is used to allocate user upload power in the third step. The effective interference is recalculated based on the updated upload power in the fourth step, and the users offloading benefits is obtained in the final step.

V. NUMERICAL RESULTS

Without loss of generality, we consider an UDN scenario with multiple MEC servers. We set the number of MECs is 10, i.e., the UDN is composed of 10 base stations, and the number of sub-channels is 10. The number of tasks in the range [10,100], and randomly distributed in the UDN. In order to evaluate the proposed algorithm HTOA, we set the background noise power $\sigma^2 = -100$dBm, bandwidth $B = 2$MHz, each of sub-channel bandwidth is $W = B/10 = 0.2$MHz, local computing resource $f_i^l = 0.1$GHz, and delay weight $w_i^t \in \{0.2, 0.4, 0.6, 0.8\}$ for UEs tasks. For the channel gain, the path fading model is adopted, defined as $g_{i,j} = L_{i,j}^{-\theta}$, where $L_{i,j}^{-\theta}$ represents the distance from offloading task T_i to offload to MEC j, and θ is the path loss parameter. In this paper, we set the parameter of θ equal to 2 [43]. The maximum of upload power $p_i^{\max} \in [0.51]$, and coefficient $\gamma = 10^{-11}$. Each experimental test is 100 and the average value is taken as the final result, so as to the accuracy and correctness of the results.

In this paper, we investigate the problem of task offloading, and the goal is mainly to optimize the delay and energy consumption required by users during the execution of tasks. Therefore, the user's offload benefit (the weighted sum of delay and energy consumption) is selected as the only metric. We proposed algorithm HTOA compares with the random offloading algorithm. The random offloading algorithm has randomness for offloading task selection, channel selection, and MEC server allocation. However, this randomness does not take into account the channel allocation of MEC computing resources.

We set that $f_j^c = 1$GHz, and $w_i = 0.1$GHz. We selected 1000, 2000 and 3000KB as the data size of task to evaluate the users offloading benefits, as shown in Figure 2. According to the overall trend of Figure 2, it can be seen that the user offloading benefits of the proposed algorithm HTOA is better than the random offloading algorithm. As the number of users increases, more users compete for limited channel resources, which cause that the user offloading benefits of the random offloading algorithm declining. And our proposed algorithm HTOA is still increasing and is significantly better than the random offloading algorithm. It can be shown that the proposed algorithm HTOA is suitable for task offloading in UDN multi-user scenarios. When the number of tasks is 10, the corresponding user offloading benefits under the three task data size are 9.37, 9.36 and 9.34, respectively, and the difference is about 0.01. When the number of tasks

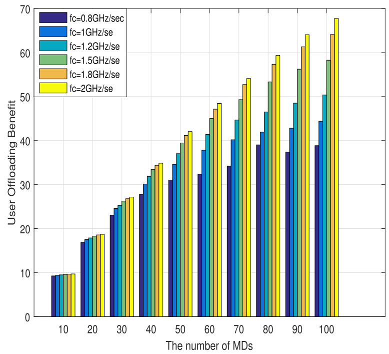


FIGURE 3. The impact of MEC computing resource for user offloading benefits.

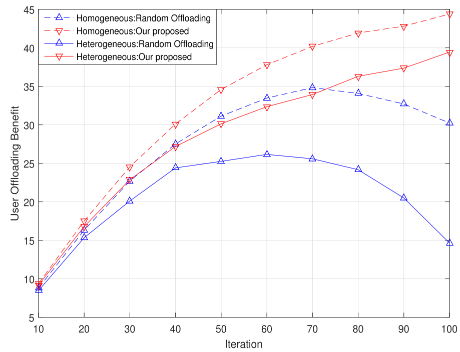


FIGURE 4. The change of user offloading benefits as the number of tasks increase in homogeneous and heterogeneous networks.

is 100, the corresponding user offloading benefits are 44.39, 42.93 and 41.54, respectively, and the difference is about 1.5. From the difference of user offloading benefits between the HTOA algorithm and random offloading algorithm, we can see that the upload delay and energy consumption of the offloading user is affected by the size of task data.

In order to analyze the impact of different MEC computing resources on the task offloading benefits, we selected six types of MEC computing capabilities, as shown in Figure 3. As we can see from the overall trend, with the increase of MEC computing resource, the benefit of task offloading benefits also increases. More specifically, in the case of a small amount of tasks, the trend of task offloading benefits under different MEC computing resources is not obvious. The gap in the task offloading benefits has gradually increased from the number 60 of tasks. The task offloading benefits increases with the increase of MEC computing resource. When the MEC's computing resource is continuously enhanced, the execution delay of the offloading task in the MEC is greatly shortened, the users offloading improving efficiency.

In order to change the impact of the task offloading benefits in homogeneous and heterogeneous networks, we set the homogeneous networks: the data size of task is 1000KB, the required computation capability of task is 0.2GHz and the local computation capability of task is 0.1GHz, and the heterogeneous networks are: the data size of task is between 1000KB and 3000KB, the required computation capability of task is between 0.1GHz and 0.4 GHz, the local computation capability of task is between 0.1GHz and 0.15GHz, as shown in Figure 4. As the number of tasks increase, the user offloading benefits of proposed algorithm HTOA increases under heterogeneous and heterogeneous networks, and the random offloading algorithm starts to decrease after the number of tasks exceeds 70. More specifically, with the increase of the amount of tasks, the task offloading benefits in the heterogeneous networks is slightly higher than that in a homogeneous environment. In addition, the reduction of random offloading algorithm is more than 1.5 times of the proposed algorithm HTOA after the number of tasks exceeds 60. It is noteworthy that it was more than 3 times when the number of task was 100. From this point of view, the proposed algorithm HTOA is more stable than the random offloading algorithm.

VI. CONCLUSION

In this paper, we propose a heuristic task offloading algorithm HTOA for investigated the joint radio resource management and task offloading in the UDN architecture. In order to maximize the task offloading benefits (the weighted sum of delay and energy consumption), we considered the management of radio resources: the MEC computing resource allocation, the channel allocation, and the user upload power control. The proposed algorithm HTOA achieves optimized task offloading by iteratively updating radio resource management. Simulation results show that the algorithm HTOA has higher user offloading benefits than the random offloading algorithm efficiently. Especially, when there are a large number of users, the advantage is more obvious, which further proves that the proposed algorithm HTOA is better applicable to UDN environment. In future work, we will further consider the complex scenario of user mobility and the energy consumption of MEC servers.

REFERENCES

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022," Cisco, San Jose, CA, USA, White Paper, Feb. 2017.

[2] G. John and T. David, "Mobile user data mining and its applications," in *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2008, pp. 1519–1538.

[3] 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.

[4] *Mobile-Edge Computing-Introductory Technical White Paper*, ETSI, Sophia Antipolis, France, Sep. 2014.

[5] IMT-2020(5G), "PG-white paper on 5G vision and requirements," China IMT-2020 Promotion Group, White Paper, May 2014.

[6] J. Sachs, G. Wikstrom, T. Dudda, R. Baldemair, and K. Kittichokechai, "5G radio network design for ultra-reliable low-latency communication," *IEEE Netw.*, vol. 32, no. 2, pp. 24–31, Mar. 2018.

[7] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 72–79, Feb. 2016.

[8] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[9] 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.

[10] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1451–1455.
[11] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, "Mobile-edge computation offloading for ultradense IoT networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4977–4988, Dec. 2018.
[12] L. Yang, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, "A distributed computation offloading strategy in small-cell networks integrated with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2762–2773, Dec. 2018.
[13] W. Chen, D. Wang, and K. Li, "Multi-user multi-task computation offloading in green mobile edge cloud computing," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 726–738, Sep. 2019.
[14] C. Bottai, C. Cicconetti, A. Morelli, M. Rosellini, and C. Vitale, "Energy-efficient user association in extremely dense small cell networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2014, pp. 1–5.
[15] 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.
[16] S. Pang, W. Zhang, T. Ma, and Q. Gao, "Ant colony optimization algorithm to dynamic energy management in cloud data center," *Math. Problems Eng.*, vol. 2017, pp. 1–10, Dec. 2017.
[17] S. Pang, Q. Gao, T. Liu, H. He, G. Xu, and K. Liang, "A behavior based trustworthy service composition discovery approach in cloud environment," *IEEE Access*, vol. 7, pp. 56492–56503, 2019.
[18] S. Pang, W. Li, H. He, Z. Shan, and X. Wang, "An EDA-GA hybrid algorithm for multi-objective task scheduling in cloud computing," *IEEE Access*, vol. 7, pp. 146379–146389, 2019.
[19] T. Song, Y. Wang, G. Li, and S. Pang, "Server consolidation energy-saving algorithm based on resource reservation and resource allocation strategy," *IEEE Access*, vol. 7, pp. 171452–171460, 2019.
[20] T. Song, S. Pang, S. Hao, A. Rodríguez-Patón, and P. Zheng, "A parallel image skeletonizing method using spiking neural p systems with weights," *Neural Process. Lett.*, vol. 50, no. 2, pp. 1485–1502, Oct. 2019.
[21] S. Pang, J. Yao, X. Wang, T. Ding, and L. Zhang, "Transmission control of MPTCP incast based on buffer balance factor allocation in data center networks," *IEEE Access*, vol. 7, pp. 183428–183434, 2019.
[22] H. He, G. Xu, S. Pang, and Z. Zhao, "AMTS: Adaptive multi-objective task scheduling strategy in cloud computing," *China Commun.*, vol. 13, no. 4, pp. 162–171, Apr. 2016.
[23] T. M. Ma, S. C. Pang, W. G. Zhang, and S. H. Hao, "Virtual machine based on genetic algorithm used in time and power oriented cloud computing task scheduling," *Intell. Autom. Soft Comput.*, vol. 25, no. 3, pp. 605–613, 2019.
[24] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
[25] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.
[26] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.
[27] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.
[28] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.
[29] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.
[30] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: Task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Trans. Cloud Comput.*, to be published.
[31] 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.
[32] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.
[33] M. Chen, Y. X. 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.
[34] H. Guo, J. Zhang, J. Liu, and H. Zhang, "Energy-aware computation offloading and transmit power allocation in ultradense IoT networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4317–4329, Jun. 2019.
[35] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12313–12325, Dec. 2018.
[36] L. Li, X. Zhang, K. Liu, F. Jiang, and J. Peng, "An energy-aware task offloading mechanism in multiuser mobile-edge cloud computing," *Mobile Inf. Syst.*, vol. 2018, pp. 1–12, Apr. 2018.
[37] Z. Ning, X. Wang, J. J. P. C. Rodrigues, and F. Xia, "Joint computation offloading, power allocation, and channel assignment for 5G-enabled traffic management systems," *IEEE Trans Ind. Informat.*, vol. 15, no. 5, pp. 3058–3067, May 2019.
[38] L. Cui, C. Xu, S. Yang, J. Z. Huang, J. Li, X. Wang, Z. Ming, and N. Lu, "Joint optimization of energy consumption and latency in mobile edge computing for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4791–4803, Jun. 2019.
[39] H. Yin and S. Alamouti, "OFDMA: A broadband wireless access technology," in *Proc. IEEE Sarnoff Symp.*, Mar. 2006, pp. 1–4.
[40] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2716–2720.
[41] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[42] P. Liu, J. Li, and H. Li, "Convex optimisation-based joint channel and power allocation scheme for orthogonal frequency division multiple access networks," *IET Commun.*, vol. 9, no. 1, pp. 28–32, 2014.
[43] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

SHANCHEN PANG received the graduation degree from the Tongji University of Computer Software and Theory, Shanghai, China, in 2008. He is currently a Professor with the China University of Petroleum, Qingdao, China. His current research interests include theory and application of Petri net, service computing, and trusted computing.

SHUYU WANG received the graduation degree from the Shandong Women's University of Computer Science and Technology, Jinan, China, in 2018. She is currently pursuing the master's degree with the China University of Petroleum, Qingdao, China. Her current research interests include mobile edge computing, cloud computing, and workflow scheduling.