Received: 9 September 2018 | Revised: 24 December 2018 | Accepted: 17 January 2019
DOI: 10.1002/dac.3928

RESEARCH ARTICLE

WILEY

Online combinatorial based mechanism for MEC network resource allocation

Xiaogang Wu | Weiheng Jiang | Yu Zhang | Wanxin Yu

College of Communication Engineering, Chongqing University, Chongqing, China

Correspondence
Weiheng Jiang, College of Communication Engineering, Chongqing University, Chongqing 400044, China.
Email: whjiang@cqu.edu.cn

Funding information
the Core Electronic Devices, High-end General Chips and Basic Software Products Projects of China, Grant/Award Number: 2017ZX01030204; Sichuan Science and Technology Program, Grant/Award Number: 2018GZ0228; Chongqing Frontier and Applied Basic Research Project, Grant/Award Number: cstc2015jcyjA40021 and cstc2017jcyjBX0047; the Fundamental Research Funds for the Central Universities, Grant/Award Number: 106112016CDJXY500002

Summary

In this paper, the resource allocation problem for mobile edge computing (MEC) network is discussed. We focus on the design of online resource allocation strategy by considering the time-varying demands of the combinatorial resources for the mobile users. To handle this issue, at first, a time splitting-based online allocation mechanism is introduced. Then, for a given resource allocation period of the time splitting, a combinatorial auction mechanism for the combination resources allocation is proposed. We further prove that the proposed combinatorial auction is both individual rational and incentive-compatible, and which can bring a higher revenue to the service provider. In order to verify the performance of the proposed mechanism, a variable-relaxed-based performance-bound algorithm and a greedy optimization-based suboptimal algorithm are presented for comparison purpose. Finally, the performance of these proposed algorithms are testified by numerical simulations, which confirm the analysis results.

KEYWORDS

combinatorial auction, joint resource allocation, MEC network, online mechanism

1 | INTRODUCTION

With the rapid developments of mobile Internet and Internet of things (IoT), more and more mobile applications, such as interactive games, augmented reality (AR), and others are deployed to mobile terminals to improve our quality of life and work efficiency.[1] At the same time, a large number of IoT devices are widely used in industry, medicine, and transportation. However, mobile terminals and especially the low-cost IoT terminals face the dilemma of limited computing resources and battery capacity, which bring a very bad experience to users. To handel these issues, recently, a new network technology has been proposed, ie, mobile edge computing (MEC).[2-4] Its main idea is that servers are deployed at the edge of the network to provide mobile users (MUs) with a highly reliable, low-latency computing and communication environment. With the MEC, the computing capability of mobile terminals could be enhanced, and by computing offloading, their energy consumption can be reduced. However, in the MEC-based system, computing offloading renders communication and computing coupled with each other, and the user-perceived performance is jointly determined by the offloading decision and the resources allocation.

Till now, lots of work have been carried out for the resources allocation problem of the MEC system. At first, the offloading decision and resources allocation depend on the properties of the task, ie, binary offloadable task[5-10] or partial offloadable task.[11-14] For the former, the tasks are either locally implemented or offloaded to the cloud to execute, while for the latter, a task can be segmented into at least two parts, and one of them is performed at local device, and the other

 | |

is offloaded to the cloud platform. In addition, offloading decision and resources allocation are affected by the system access scheme, which determines how MUs share both the communication and computing resources in the system. For the communication resources, both orthogonal access schemes, ie, time-division multiple access (TDMA)[11] and orthogonal frequency division multiple access (OFDMA),[8,11,15,16] and nonorthogonal access schemes, ie, code division multiple access (CDMA)[17,18] and nonorthogonal multiple access (NOMA),[14,19] have been discussed yet. For the computing resource, depending on how many virtual machines cloud be virtualized by the system, multiple tasks could be serially or concurrently executed at the cloud. Furthermore, the system design objective or the user-perceived performance metric also plays a leading role in the offloading decision and resources allocation. Typically, system design objectives are latency minimization,[12] energy consumption reduction,[12,13,15,16] and delay-energy tradeoff.[5-10]

From a methodological point of view, the most aforementioned work of the resources allocation problems are modeled by the optimization theory. However, as an important branch of game theory,[20] the auction theory is widely used in the field of wireless network resource allocation recently, such as cognitive network,[21] HetNets,[22] and the mobile cloud networks.[23-25] In Zhang et al,[24] the authors proposed a multi-round-sealed sequential combinatorial auction mechanism for the communication and computing resource allocation in the MEC network. However, there is no difference between delay-sensitive users and ordinary users, and the quality of service (QoS) of delay-sensitive users may not be guaranteed. Mashayekhy et al[26] designed an auction-based online mechanism for virtual machine (VM) provisioning, allocation, and pricing in clouds that consider several types of resource. This mechanism is used in traditional cloud computing, only considering the online allocation of computing resources but not considering the joint allocation of communication and computing resources.

By analyzing the existing works, we note that most of these studies only focus on the static network setting, ie, user's requirements are given at the beginning and remain unchanged over the whole resources allocation period. However, in practice, because of the mobility of users or the dynamics of their demands, the resources requirements are time-varying. In addition, most of the proposed auction mechanisms aimed at maximizing the social welfare,[20] which causes low revenues to the buyers(service providers). In this paper, the problem of resources allocation for MEC is discussed, and our focus is on the online allocation strategy design under time-varying resource requirements. In specific, the online allocation mechanism is introduced at the beginning. Then, the problem of resources allocation within a fixed period of time is modeled based on the combinatorial auction, and an allocation mechanism that satisfies individual rationality and incentive compatibility is proposed. In addition, for the sake of comparison and analysis, a performance-bound algorithm based on variable relaxation and a suboptimal algorithm based on greedy optimization are proposed for the resource allocation problem in a fixed period. Finally, the performance of the proposed algorithms are verified by simulations.

The organization of the remainder of the paper is as follows. In Section 2, we present the network model and introduce an online combinatorial resource auction problem. In Section 3, based on the optimization theory, the combinatorial resource allocation problem in a fixed period is established, and a performance-bound algorithm based on variable relaxation and a suboptimal algorithm based on greedy optimization are proposed therein. Then, a resource allocation mechanism based on combinatorial auction is proposed in Section 4, and we also prove the economic properties of the proposed combinatorial auction mechanism in this section. The performance of the three proposed algorithms are analyzed by numerical simulations in Section 5, and we conclude in Section 6.

2 | SYSTEM MODEL

2.1 | Network model

In this paper, we consider a single-cell MEC network, which include N (MUs), a wireless access base station (BS), and a MEC server, as shown in Figure 1. The MUs are denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$, and all of them have delay-sensitive and computation-intensive tasks to perform. Because of their limited computing capability and energy supply, they cannot locally perform these tasks and have to offload them to the MEC server. We define the communication resources provided by the network as a wireless access channel at the BS with bandwidth W Hz. The computing resources is the CPU at the MEC server with frequency C Hz. Here, an OFDMA access scheme is adopted at the BS, ie, multiple MUs access to the BS on orthogonal frequency bands, and the CPU resource of the MEC server is virtualized to multiple VMs and allocated to different MUs concurrently. It is different from existing research that assumed all the MUs arrived at the same time[24,27]; herein, we assume that MUs' demands are time-varying and that the MUs arrive at the system dynamically. We design a multi-user online combinatorial resources allocation mechanism to optimize the utility, ie, the seller's revenue.

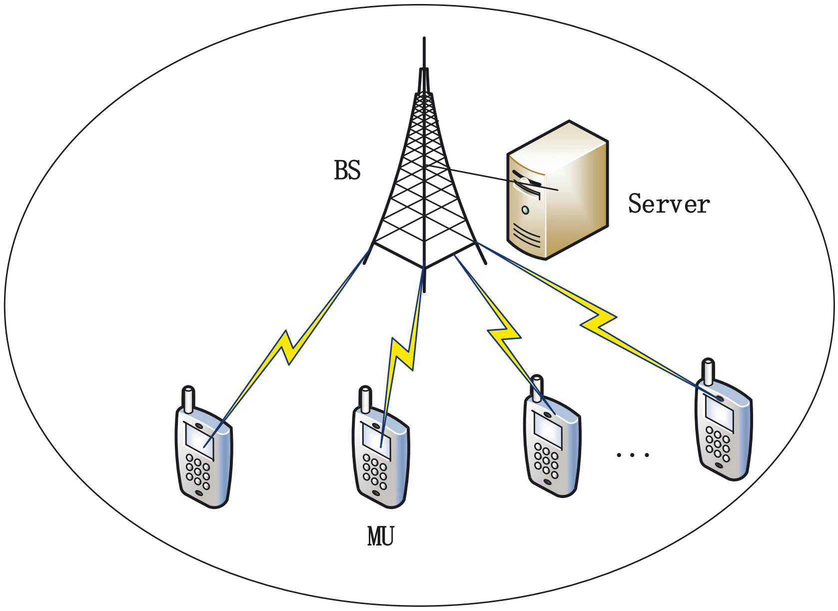

FIGURE 1 The network model

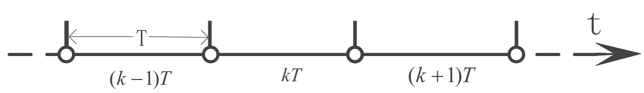

FIGURE 2 The process of the online mechanism

2.2 | Online mechanism and auction model

Because the MUs' resource demands arrive at the system dynamically, and the MUs ask for combinatorial resources, the resources allocation mechanism includes both the online mechanism and the combinatorial resources allocation mechanism, as follows.

2.2.1 | Online mechanism based on a fixed period *T*

The idea of the online mechanism is based on a fixed period T as shown in Figure 2, ie, the resources allocation is performed over each fixed period T. In specific, the system collects the requests/bids information for the combinatorial resource of the kth period in the $k-1$th period and implements the resource allocation at the end of the $k-1$th period. Then, based on the allocation result, the winning MUs finish the payments and obtain the resources in kth period. The failed users or the newly arrived users will start bidding for the resources of the $k+1$th period in the kth period, and the whole process is repeated.

2.2.2 | Combinatorial auction model

On the basis of the above online mechanism, the multi-user online combinatorial resource competition is then modeled as a combinatorial auction within a period T. Herein, the BS is the seller and model as the auctioneer, the computing and communication resources are the auction items, and the MUs are the bidders. Specifically, define the bid of user i as $\theta_i = \left(w_i, c_i, t_i, b_i^t, b_i^s\right)$, in which w_i and c_i denote the communication and computing resources demands of MU i, respectively, t_i is the amount of time for which the requested resources must be allocated (here $t_i \le T$), the bid for service time priority of MU i is denoted by b_i^t, hereinafter referred to as time bid, and the bid for combinatorial resource is denoted by b_i^s, hereinafter referred to as resource bid. The bids of all MUs are characterized by $\Theta = \left(\theta_1, \theta_2, \dots, \theta_{|\mathcal{N}|}\right)$. In addition, the real demand information of user i is $\bar{\theta}_i = \left(\bar{w}_i, \bar{c}_i, \bar{t}_i, \bar{b}_i^t, \bar{b}_i^s\right)$, ie, the true resources demand and valuation of MU i.

2.3 | Problem model

On the basis of the above auction model, we define x_i to indicate the resources allocation result, ie, $x_i = 1$ if user i is allocated with the requested resources and $x_i = 0$ otherwise. The value function of user i is v_i^s defined as[26]

$$v_i^s = \begin{cases} \bar{b}_i^s & \text{if } x_i = 1, \\ 0 & \text{if } x_i = 0. \end{cases} \tag{1}$$

v_i^s is the real valuation of user i when the resource is allocated. Define the payment of MU i as P_i, which indicates both time priority fees P_i^t and the resource usage fees P_i^s. P_i^t and P_i^t are determined by the payment rule. In specific, the payment of user i is denoted as follows

$$P_i = \begin{cases} P_i^t + P_i^s & \text{if } x_i = 1, \\ 0 & \text{if } x_i = 0. \end{cases} \tag{2}$$

Thus, the total utility U_i of user i is expressed as[20]

$$U_i = \begin{cases} U_i^t + U_i^s = b_i^t + b_i^s - \left(P_i^t + P_i^s\right) & \text{if } x_i = 1, \\ 0 & \text{if } x_i = 0, \end{cases} \tag{3}$$

where U_i^s is resources bid utility, and U_i^t is time priority bid utility.

With the above, the details of the resources allocation mechanism, which determine the competition among users, the winners, and the fees paid by the winners, are discussed in the following sections.

3 | OPTIMIZATION PROBLEM AND ALGORITHM

In order to evaluate the proposed combinatorial auction mechanism, optimization-based combined resource allocation problem is formulated herein, and two benchmark algorithms are presented. Specifically, an optimization problem is established for the combinatorial resources allocation in one period T. Since the formulated problem is a mixed integer programming problem, it is hard to solve directly. Therefore, a performance-bound algorithm based on variable relaxation (VBA) and a heuristic algorithm based on greed thought (GOA) are proposed.

3.1 | Optimization problem model

Herein, we discuss the combinatorial resources allocation for a period T by optimization theory.[28] We denote the time of period T as the set $\mathcal{T} = \{t|0 \le t \le T\}$. As shown in Figure 3A, in fact, the resources allocation in a period T can be modeled as a multiple dimension knapsack problem (MDKP),[29] that is, filling the three-dimensional space by the cubes as much as possible. While one cube represents one resources request from one MU, and the boundary denotes the constraint on the available resource in the system, ie, T denotes the allocated time period, W denotes the available communication resource, and C denotes the available computing resources. In addition, for the resources allocation, there are some other restrictions, ie, the cube filling cannot overlap in the C-dimension and the W-dimension, but can overlap in the T-dimension, as shown in Figure 3B. This comes from the fact that both the communication and computing resources

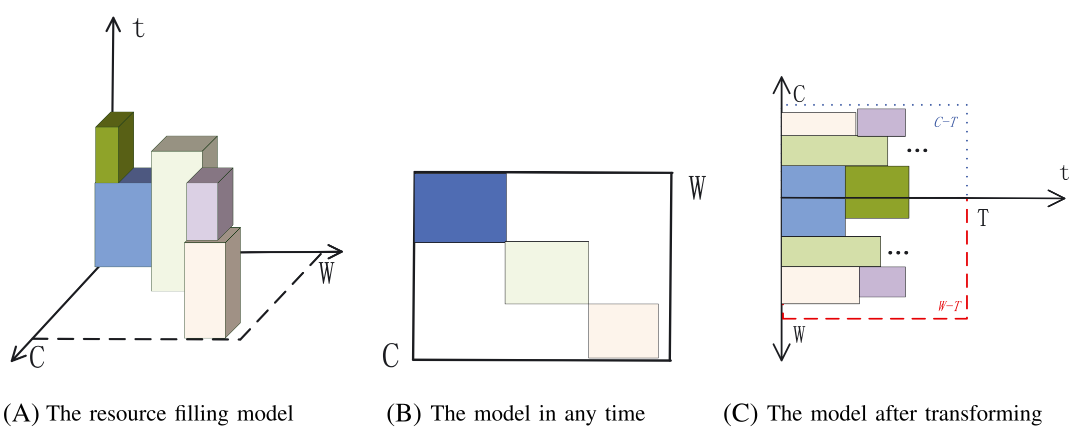

FIGURE 3 The resource allocation model in kth period

are exclusively allocated to the MUs at a certain time. Through the analysis of the filling model, and based on Figure 3A,B, the three-dimensional filling model can be transformed into a filling model of two rectangles in a two-dimensional plane, ie, the $C - T$ plane and $W - T$ plane in Figure 3C, then it can be solved by the optimization theory. The goal of the optimization problem here is to maximize the social welfare V. Therefore, we have the optimization problem IP1 as below.

$$IP1 : \max V = \sum_{i\in\mathcal{N}} \sum_{t\in\mathcal{T}} v_i^s = \sum_{i\in\mathcal{N}} \sum_{t\in\mathcal{T}} x_{it} b_i^s \tag{4}$$

$$s.t. \sum_{t\in\mathcal{T}} x_{it} \leq 1, \forall i \in \mathcal{N}, \tag{5}$$

$$\sum_{i\in\mathcal{N}} x_{it} c_i \leq C, \forall t \in \mathcal{T}, \tag{6}$$

$$\sum_{i\in\mathcal{N}} x_{it} w_i \leq W, \forall t \in \mathcal{T}, \tag{7}$$

$$x_{it} \in \{0, 1\}, \forall i \in \mathcal{N}, t \in \mathcal{T}. \tag{8}$$

Herein, x_{it} is used to indicate whether MU i is assigned resource at t or not. The objective function (4) maximizes the social welfare $\sum_{i=1}^{|\mathcal{N}|} \sum_{t=0}^{T} v_i^s$. In IP1, (5) denotes that the resource request of each MU at most can be satisfied once in one period. (6) and (7) restrict the total amount of computing and communication resource in the system, respectively. (8) indicates that the decision variable takes values of 0 or 1.

By observation, we know that IP1 is an integer programming. In addition, the number of decision variables in IP1 is infinite. These make IP1 very hard to address on. The authors in Mashayekhy et al [26] solve a simpler IP problem by a CPLEX solver, but we find that this method is hard to solve IP1. Furthermore, in the following two subsections, a performance-bound algorithm based on variable relaxation and a centralized greedy optimization algorithm are proposed respectively.

3.2 | A performance-bound algorithm based on variable relaxation

On the basis of variable relaxation, a performance-boundary algorithm is proposed herein. The main idea is that the original filling problem is relaxed as follows. Only the area of filling area, ie, CT and WT, but not the filling boundary, ie, the rectangle $C - T$ and $W - T$, in Figure 3C is limited. Obviously, this relaxation will expand the feasible domain of the original problem. Thus, the system utility obtained from the relaxation problem IP2 below can be regarded as the performance boundary of the original optimization problem IP1.

$$IP2 : \max V = \sum_{i\in\mathcal{N}} v_i^s = \sum_{i\in\mathcal{N}} b_i^s x_i \tag{9}$$

$$s.t \sum_{i\in\mathcal{N}} w_i t_i x_i \leq WT, \tag{10}$$

$$\sum_{i\in\mathcal{N}} c_i t_i x_i \leq CT, \tag{11}$$

$$x_i \in \{0, 1\}, \forall i \in \mathcal{N}, \tag{12}$$

in which the decision variable x_{it} in IP1 is degenerated to x_i in IP2. $x_i = 1$ if MU i is allocated the resource, and $x_i = 0$ otherwise. Now, (10) and (11) are the constraints of the fill area (Figure 3C). The others are the same as IP1. Then, IP2 is an integer programming problem with finite variables that can be solved by the CVX or the built-in solver of MATLAB.[30] In this paper, IP2 is solved by the built-in solver of MATLAB, and the formulated algorithm is named as variable-relaxed-based algorithm (VBA).

3.3 | A centralized greedy optimization algorithm

Herein, a heuristic optimization algorithm[31] based on greedy though and named GOA is proposed for IP1. The main idea is that based on the MUs' bidding index function $f_i = b_i^s/(t_i(w_i+c_i))$, the combinatorial resources are allocated. In specific, the larger the value of the index function, the higher the priority of the allocation. The whole process is repeated until all the available resources has been allocated. The details of the GOA algorithm are summarized in the Algorithm 1 as below.

Algorithm 1 Greedy-based optimization algorithm: GOA ($\mathcal{N}, \Theta$)

1: Initialize $W, C, T, \mathcal{N}$ and $\boldsymbol{a} = \mathbf{0}$;
2: Collect bids from all users, ie, $\Theta = (\theta_1, \theta_2, \dots, \theta_{|\mathcal{N}|})$ and $\theta_i = (w_i, c_i, t_i, b_i^t, b_i^s)$, the resource requested by each MU i is converted to rectangle with area $w_i t_i$ and $c_i t_i$;
3: For $\forall i \in \mathcal{N}$, calculate the bidding index function $f_i = b_i^s/(t_i(w_i + c_i))$;
4: Sort all users according to f_i, and we have $f_{[1]} \geq \dots \geq f_{[k]} \geq, \dots \geq f_{[|\mathcal{N}|]}$;
5: For $\forall i = [1] \dots [|\mathcal{N}|]$, implements the following steps:
6: Fill the rectangles CT and WT by $w_i t_i$ and $c_i t_i$, respectively, with the principle of filling the portions near the C-axis and the W-axis in Figure 3C first, and the $w_i t_i$ and $c_i t_i$ are paired and located separately over the two sides of the T-axis;
7: If $w_i t_i$ and $c_i t_i$ can be filled in $C - T$ and $W - T$, the allocation is successful, and let $a_i = 1$; otherwise, $a_i = 0$, and we continue this process for the next MU;
8: End and output $\boldsymbol{a}$.

In Algorithm 1, $\boldsymbol{a} = \left(a_1, \dots, a_i, \dots, a_{|\mathcal{N}|}\right)$ is a vector with $|\mathcal{N}|$ dimensions, which indicates the resource allocation result for each MU. If MU i is allocated with the resource, $a_i = 1$; otherwise, $a_i = 0$. Algorithm 1 includes three phases. First, the system collects the bids $\Theta = \left(\theta_1, \theta_2, \dots, \theta_{|\mathcal{N}|}\right)$, where $\theta_i = \left(w_i, c_i, t_i, b_i^t, b_i^s\right)$. Then, according to the descending order of the index function, the two rectangles $W - T$ and $C - T$ are filled in order. The principle of the filling is that filling the portions near the C-axis and the W-axis in Figure 3C first, and the $w_i t_i$ and $c_i t_i$ are paired and located separately over the two sides of the T-axis. Finally, the algorithm outputs the result.

4 | ONLINE COMBINATORIAL AUCTION MECHANISM

According to the online mechanism presented in Section 2, we propose an online combinational auction–based algorithm (OCAA) for the kth period, as shown in Algorithm 2 below. It includes five steps, ie, collection of bids, user grouping, time priority bidding between groups, resource bidding within the group, and payment determination.

Algorithm 2 Online combinatorial auction mechanism: OCAA ($\mathcal{N}, \Theta$)

1: Initialize $W, C, T, \mathcal{N}, \boldsymbol{a}^G, \boldsymbol{p}^t, \boldsymbol{a}^S$, and $\boldsymbol{p}^S$;
2: Collect bids from all users, ie, $\Theta = (\theta_1, \theta_2, \dots, \theta_{|\mathcal{N}|})$;
3: User grouping and time priority bidding between groups: $(\boldsymbol{a}^G, \boldsymbol{p}^t) \leftarrow \text{Group}(\mathcal{N}, \Theta)$;
4: Resource bidding within the group: $\boldsymbol{a}^S \leftarrow \text{Alloc}\ (\mathcal{N}, \Theta, \boldsymbol{a}^G)$;
5: Payment determination: $\boldsymbol{p}^S \leftarrow \text{Pay}\ (\mathcal{N}, \Theta, \boldsymbol{a}^S)$;
6: End and output $(\boldsymbol{a}^G, \boldsymbol{p}^t, \boldsymbol{a}^S, \boldsymbol{p}^S)$.

In Algorithm 2, the user grouping vector $\boldsymbol{a}^G = \left(a_1^G, a_2^G, \dots, a_N^G\right)$ characterizes which group the users belong to, and $a_i^G = 0$ means that the user i fails in grouping. The user time priority payment vector $\boldsymbol{p}^t$ indicates the required payments under current group. The user resources allocation vector $\boldsymbol{a}^S$ indicates the resources allocation results. The element is 1 if the corresponding MU is allocated with the required resource, otherwise it takes value 0. The user resources bid payment vector $\boldsymbol{p}^S$ represents the fee that the MU has to pay for the allocated resources.

The execution of the OCAA is as follows. First, the BS collects the bid information. Second, the users are grouped according to t_i (service type). Third, the OCAA mechanism performs time priority bidding to determine time segmentation of each group according to the sum of time bids, ie, steps 2 and 3. Subsequently, the mechanism allocates the resources for each user group and determines the payment of each user. Finally, the algorithm outputs all the results. Next, we will depict the steps of OCAA in details.

4.1 | The bid information collecting

The system collects bidding requests in the $k - 1$th period (including the new users arrived in the $k - 1$th period and the losers before the $k - 1$th period). At the end of the $k - 1$th period, we perform resource allocation and determine the winners. For MU i, its bid information is characterized by $\theta_i = \left(w_i, c_i, t_i, b_i^t, b_i^s\right)$. Each user has a unique user ID, and it is allowed to bid at most once in one resources allocation period $\mathcal{T}$. The sealed bid is used, ie, the bid information is private information, and each user does not know the bid information of the remaining users. We assume that the overhead of signal interaction in the bid information collecting can be ignored.[26]

4.2 | User grouping

User grouping stems from the fact that similar applications have similar request time in resources use and also have similar delay sensitivity, while the resource occupation time of significant different applications are generally different and have different delay limits. Thus, user grouping can distinguish the delay limit level for different users. Therefore, based on the requested resource occupation time, Algorithm 3 is proposed to group the users and to determine the payment of each user. The basic idea is grouping the users according to their request resources usage time. In the following, $\mathcal{G}_m$ represents the mth group, the user resources occupation time in the same group is approximately equal, and the total number of groups is M (in the considered allocation period). In addition, grouping satisfies the following equations:

$$\bigcup_{m=1}^{M} \mathcal{G}_m = \mathcal{N}, \bigcap_{m=1}^{M} \mathcal{G}_m = \emptyset. \tag{13}$$

Herein, $\mathcal{N}$ represents the user set asked resources in the considered period. In essence, grouping is a time division of the resource in one period, ie, the resource is divided into unequal time segments, and users within equal segments are allocated resources in the same time period.

Algorithm 3 User grouping: Group $(\mathcal{N}, \Theta)$

Initialize T, $\mathcal{N}$, $\boldsymbol{a}^G$, and $\boldsymbol{p}^t$;
Collect bids from all users, ie, $\Theta = (\theta_1, \theta_2, \dots, \theta_{|\mathcal{N}|})$;
For $\forall i \in \mathcal{N}$
    The users that have the same request time slice t_i are grouped in the same group $\mathcal{G}_m$, and promise that $\bigcup_{m=1}^{M} \mathcal{G}_m = \mathcal{N}$, $\bigcap_{m=1}^{M} \mathcal{G}_m = \emptyset$;
Count the total number of groups M, initialize $\boldsymbol{t}^G = \mathbf{0}$;
For $m = 1, \dots, M$
    Calculate the sum of time bidding of group $\mathcal{G}_m$: $S_m^t = \sum_{i\in\mathcal{G}_m} b_i^t$;
    The value of t_m^G is the time slice requested by the user in group $\mathcal{G}_m$;
Sort all groups in descending order of S_m^t, make $S_{[1]}^t \geq, \dots, \geq S_{[k]}^t \geq, \dots, \geq S_{[M]}^t$;
If $sum(\boldsymbol{t}^G) > T$
    The successful groups are the n groups that satisfy $\sum_{k=1}^{k=n} t_{[k]} \leq T$ and $\sum_{k=1}^{k=n+1} t_{[k]} \geq T$, the number of successful groups is n;
    All users in the successful group have $a_i^G = m$, otherwise $a_i^G = 0$;
Else
    All groups are successful, the number of successful group is $n = M$;
    For all users in the winning group, have $a_i^G = m$;
For all successful groups, $m = 1, \dots, n$
    The user that has the lowest time bid is $p = \underset{i\in\mathcal{G}_m}{arg} \min b_i^t$;
    Eliminate the user p and let $a_p^G = 0$, then user p pays $p_p^t = 0$;
    For all users $i \in \mathcal{G}_m \wedge i \neq p$, we have $p_i^t = b_p^t$;
End and output $(\boldsymbol{a}^G, \boldsymbol{p}^t)$.

4.3 | The time priority bidding between groups

On the basis of the result of user grouping, time priority of each group and MU is determined. Specifically, the time priority of each group depends on the sum of time bidding of the users in the group, ie,

$$S_m^t = \sum_{i\in\mathcal{G}_m} b_i^t, \tag{14}$$

where S_m^t is the sum of time bids of users in group m. Then, the time priority of each group in resources allocation is determined by S_m^t, ie, we sort $S_m^t, m = 1, \dots, M$ in a descending order. The group with larger value of S_m^t will be allocated resources earlier. The allocation process is ended until the total time period T is completely allocated. $\boldsymbol{t}^G = \left(t_1^G, \dots, t_m^G, \dots, t_M^G\right)$ denotes the time slice of each group. The details of the grouping are described in step 6 to step 15 of Algorithm 3.

On the basis of the above analysis, the time priority payment rule is defined as follows. For any winning group in the kth period, defining the lowest time bid user of this group is p, and the corresponding time bid is b_p^t. User p will be rejected and will no longer participate in the resources allocation; its time priority payment is 0. The rest of the users in this group will pay the time bid b_p^t of user p. All members in the failed group of the kth period will pay 0. These ideas are characterized by step 16 to step 19 of Algorithm 3.

4.4 | The resource bidding within the group

On the basis of the result of user grouping, resources bidding within the group is performed, and Algorithm 4 illustrates the resource bidding process in a group. For any winning group $\mathcal{G}_m, m = 1, .., n$, the resource allocation result depends on the user's bid density. For MU i, its resources bid density is defined as[26]

$$d_i = \frac{b_i^s}{t_i \times w_i \times c_i}. \tag{15}$$

One can note that the resources bid density essentially depicts the user bid price of unit resource cube (communication, computing, and time). Obviously, the greater the bid density, the greater this user valuated the unit resource cube. On that basis, the resources allocation in any group is determined by the bid density as follows.

Algorithm 4 Resource allocation within the group: Alloc $(\mathcal{N}, \Theta)$

1: Initialize C, W, $\mathcal{N}$, $\boldsymbol{a}^S$, and s;
2: For all groups won in the bidding, ie, $m = 1, \dots, n$
3: Recover each group's resource C, W;
4: For all users $i \in \{1, \dots, |\mathcal{G}_m| - 1\}$, calculate the resource bid density, ie, $d_i = b_i^s/(t_i \times c_i \times w_i)$;
5: Sort all users in group m according to d_i, then we have $d_{[1]} \geq, \dots, \geq d_{[k]} \geq, \dots, \geq d_{[|\mathcal{G}_m|-1]}$;
6: For $i = [1], \dots, [|\mathcal{G}_m| - 1]$
7: If $b_i^s > s$, then $W = W - w_i$, $C = C - c_i$ and $a_i^s = 1$; otherwise, $a_i^s = 0$;
8: If $W \leq \underset{i \in \mathcal{G}_m \backslash p}{min}\ w_i \vee C \leq \underset{i \in \mathcal{G}_m \backslash p}{min}\ c_i \vee \underset{i \in \mathcal{G}_m \backslash p}{sum}\ a_i^s = |\mathcal{G}_m| - 1$, the resource allocation process is completed, we set $a_i^s = 0$ and go to step 2; otherwise, go to step 6;
9: End and output $\boldsymbol{a}^s$.

Herein, s is the resource reserve price, which is used to promise the desired incentive compatibility, ie, if the resources bid of MU i satisfies $b_i^s < s$, MU i will be rejected to allocate any resources. From Algorithm 4, we note that the rule of the resources allocation in each winning group is the order of the resource bid density. The allocation process is ended until the available resources in the corresponding time slice have been completely occupied. The failed users can participate the resources bidding in the next allocation period or exit the system directly.

4.5 | The payment rule

The payment rule determines the resource using price of each user in each group, because the payment rule of the classical VCG mechanism cannot bring a high revenue to the service provider, therefore we design a payment rule based on critical price payment, which is summarized in Algorithm 5. There are two cases in determination of the payment for a user.

C1 (Critical price payment[32]): First, user i is removed from the system, and its allocated resources are released, then the allocation algorithm is re-executed. At this time, the algorithm may find a failed user q that has not been allocated resources in the previous round of resources allocation (including user i) but can obtain resource in current round of resources allocation without user i. It is obvious that the resource bid density of user i and q satisfy $d_i > d_q$, ie, the minimum resource bid density required for user i to obtain resource is d_q, thus its resource payment is $d_q \times (w_i \times c_i \times t_i)$.

C2 (Reserve price payment): If the user q in C1 cannot be found, user i will pay with the reserved price s.

Algorithm 5 Payment rule: Pay $(\mathcal{N}, \Theta, \boldsymbol{a}^s)$

1: Initialize C, W, $\mathcal{N}$, $\boldsymbol{p}^S = \mathbf{0}$ and s;
2: For all groups won in the bidding, ie, $m = 1, \dots, n$
3: For all wining users $i \in \mathcal{G}_m \wedge i \neq p \wedge a_i^s = 1$
4: $\bar{\boldsymbol{a}}_s \leftarrow Alloc(\mathcal{N}_{-i}, \Theta_{-i}, \boldsymbol{a}^G)$
5: The critical user q is the user which satisfies $a_i^s = 0 \wedge \bar{a}_i^s = 1$, if there are more than one user, the q is the user which has the largest resource bid, and user i pays $p_i^s = d_q \times (w_i \times c_i \times t_i)$; If there is not a user q, user i's payment is $p_i^s = s$;
6: End, and output $\boldsymbol{p}^s$.

4.6 | The economic property of OCAA

This section analyzes the economic properties of the proposed online combinatorial resource auction mechanism OCAA, such as the individual rationality and incentive compatibility. The conclusions are as follows.

Property 1. (Individual rationality of OCAA). In our OCAA-based system, if the users bid truthfully, we have $U_i \geq 0$.

Proof. In order to prove Property 1, we first present Lemma 1 below, and its proof is given in the following. □

Lemma 1. *For OCAA, if the resource bidding and time priority bidding both satisfy the individual rationality, OCAA satisfies the individual rationality.*

Proof of Lemma 1. It is obvious that if $U_i^s \geq 0$ and $U_i^t \geq 0$, we have $U_i = U_i^s + U_i^t \geq 0$.

Thus, we will prove the individual rationality of resource bidding and time priority bidding separately.

The individual rationality of time priority bidding: Taking user i and its group $\mathcal{G}_m$ as examples, there are two cases for its bidding.

TB-1: The group $\mathcal{G}_m$ of user i loses in the time priority bidding, then all the users will not be allocated resource, and they will pay zero, so the utility of user i is $U_i^t = 0, \forall i \in \mathcal{G}_m$;

TB-2: The group $\mathcal{G}_m$ of user i wins in the time priority bidding, then time priority fee of user i is $p_i^t = b_p^t$ (user p is the lowest time bid user in group $\mathcal{G}_m$ and it loses in the time bidding). Obviously, $b_p^t < b_i^t$, then the utility of user i is $U_i^t = b_i^t - P_i^t = b_i^t - b_p^t > 0$, where $U_p^t = 0$. That is, we always have $U_i^t \geq 0, \forall i \in \mathcal{G}_m$ in time bidding.

The individual rationality of resource bidding: Similarly, taking user i and its group $\mathcal{G}_m$ as examples, there are three cases for its bidding.

RB-1: The group $\mathcal{G}_m$ of user i loses in the time priority bidding, then all users in this group will not be allocated resource and the payment is 0, hence the utility of user i is $U_i^s = 0$;

RB-2: The group $\mathcal{G}_m$ of user i wins in the time priority bidding but user i loses in the resource bidding, then user i will not be allocated resource, and user payment is 0, and also its utility is $U_i^s = 0$.

RB-3: The group $\mathcal{G}_m$ of user i wins in the time priority bidding, and user i wins in the resource bidding, then the payment of user i is $P_i^s = d_q \times w_i c_i t_i \leq d_i \times w_i c_i t_i = b_i^s$ and we have $U_i^s = b_i^s - P_i^s = b_i^s - b_q^s \geq 0$. That is, we always have $U_i^s \geq 0, \forall i \in \mathcal{G}_m$ in resource bidding.

In summary, according to Lemma 1, we conclude that OCAA satisfies the individual rationality. □

Property 2. In our OCAA-based system, the users that bid truthfully will obtain the highest revenue, ie, $U_i(\bar{\Theta}_i, \Theta_{-i}) \geq U_i(\Theta_i, \Theta_{-i})$.

Proof. Since the OCAA belongs to the sealed auction, ie, the number of bidders and their bid information are unknown to each other, then the time bidding and resource bidding are independent with each other. Therefore, in order to prove the incentive compatibility of the OCAA, it is equivalent to prove the time bidding and the resource bidding are both incentive compatible separately.

For the time bidding, user i bids truthfully as $\bar{b}_i^t$ and untruthfully as b_i^t, there are four cases:

C1: User i is a winner regardless of bidding truthfully or untruthfully. According to Algorithm 3, user i pays for the same price $p_i^t = b_p^t$, and the utility is the same, ie, $\bar{U}_i^t = U_i^t$;

C2: User i is a winner when bidding truthfully but is a loser when bidding untruthfully. According to individual rationality of the OCAA, bidding truthfully will ensure a positive utility, ie, $\bar{U}_i^t \geq 0$. In the later case, the utility is zero, and we have $\bar{U}_i^t \geq U_i^t$;

C3: User i is a loser regardless of bidding truthfully or untruthfully. Then, the utility is both zero, ie, $\bar{U}_i^t = U_i^t = 0$;

C4: User i is a loser when bidding truthfully but is a winner when bidding untruthfully. In this case, obviously, $\bar{U}_i^t = 0$ and $\bar{b}_i^t \leq b_p^t$. However, if the user is a winner when bidding untruthfully, user i pays for b_p^t, but the obtained utility is $U_i^t = \bar{b}_i^t - b_p^t \leq 0$, ie, $\bar{U}_i^t \geq U_i^t$.

That is, in the time bidding, the mechanism satisfies the incentive compatibility. Now, for the resource bidding, we first prove that the allocation rule of OCAA satisfies the monotonicity, and the payment rule of OCAA satisfies the critical price payment as follows.

We first show that $Alloc(\mathcal{N}, \Theta)$ is monotone. If user i is allocated with the communication resources w_i and the computing resources c_i by quoting b_i^s, it is obvious that the quotation is higher, ie, $\hat{b}_i^s > b_i^s$, or the number of resource requested by the user is less, ie, $\hat{w}_i < w_i, \hat{c}_i < c_i$, and the resources are surely obtained. Therefore, the resources allocation rule satisfies the monotonicity within a fixed period.

We now prove that the payment rule is the critical payment. In doing so, we need show that P_i^s determined by the payment rule is the minimum value that user i must report to obtain the allocation. The payment of user i is $P_i^s = d_q \times w_i c_i t_i$, and the resource bid density of user q is $d_q < d_i$, thus user q will win if i does not participate in the auction. If the quotation of i is $\hat{b}_i^s < P_i^s$, then the resources bid density of i is

$$d_i = \frac{\hat{b}_i^s}{t_i \times w_i \times c_i} < \frac{P_i^s}{t_i \times w_i \times c_i} = \frac{d_q \times t_i \times w_i \times c_i}{t_i \times w_i \times c_i} = d_q. \tag{16}$$

Equation 16 shows that the bid density of user i will be lower than user q if user i reports a price lower than P_i^s, then he/she cannot obtain the resource. Therefore, the payment of user i is the lowest quotation required for obtaining the resources, ie, the payment rule satisfies the critical payment rule.

Since the allocation rule is monotone and the payment rule is the critical payment, then the OCAA mechanism satisfies the incentive compatibility in the resources bidding.[32] To sum up, OCAA is incentive-compatible. □

5 | PERFORMANCE EVALUATION

In order to verify the performance of the proposed resource allocation algorithms in this paper, a simulation platform has been built. The performance such as social welfare, resource utilization, and user payments are evaluated.

5.1 | Simulation setup

The simulation scenarios adopted herein are as follows.[26,33]

Scenario 1. (small-capacity network). In this scenario, the available resources is insufficient. Specifically, the computing capacity C and communication bandwidth W are set to 20 and 15, respectively, ie, $W = 15$, $C = 20$, and $T = 14$.

Scenario 2. In this scenario, the available resources is sufficient. Specifically, the computing capacity C and communication bandwidth W are set to 80 and 75, respectively; still we set $T = 14$.

In both network scenarios, the requested computing resources c_i and communication resources w_i by each user follow an independent uniform distribution in the range $[3, 5]$, the resources occupation time requested by each user follows an independent uniform distribution in the range $[2, 4]$, and the number of users varies in step 5 within the interval of $[20, 55]$. Otherwise, the resource bid and time bid follow an independent uniform distribution in the range $[3, 5]$ and an independent uniform distribution in the range $[2, 4]$, respectively.[24]

5.2 | Analysis of results

First, we analyze the social welfare performance of the proposed three algorithms, ie, the OCAA, GOA, and VBA, in a small-capacity network by varying the number of users in the system. The result is shown in Figure 4. One can note that, as the number of users increases, the social welfare of these algorithms are all increased gradually. This comes from the fact that when the available resource is insufficient, the more the users are, the easier the resources is allocated to the high-value users. Thus it results to a larger social welfare. In addition, both the optimization-based algorithm GOA and VBA are superior to the auction algorithm OCAA. This can be explained as follows. Both VBA and GOA aim to maximize social welfare. While for OCAA, in order to promise the desired economic properties of the auction, it will incur some social welfare loss. Furthermore, it can be seen that although GOA is an approximation algorithm, it has a little gap with the performance boundary algorithm VBA in the social welfare. This confirms that the centralized algorithm GOA is asymptotically optimal.

Then, we analyze the resources utilization of the proposed three algorithms in a small-capacity network, and the result is shown in Figure 5. We note that as the number of users increases, the resources utilization of these algorithms rose up gradually. This is because of the fact that when the number of users increases and with different resources demands, the diversity gain promises the improvement in resources utilization. In addition, we can observe that the resources utilization rate of algorithm VBA is the highest and approaches about 98%, while the auction-based algorithm OCAA is

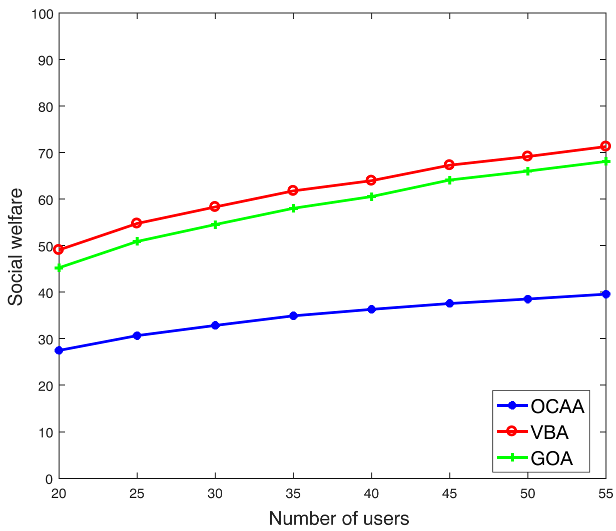

FIGURE 4 The social welfare in a small-capacity network

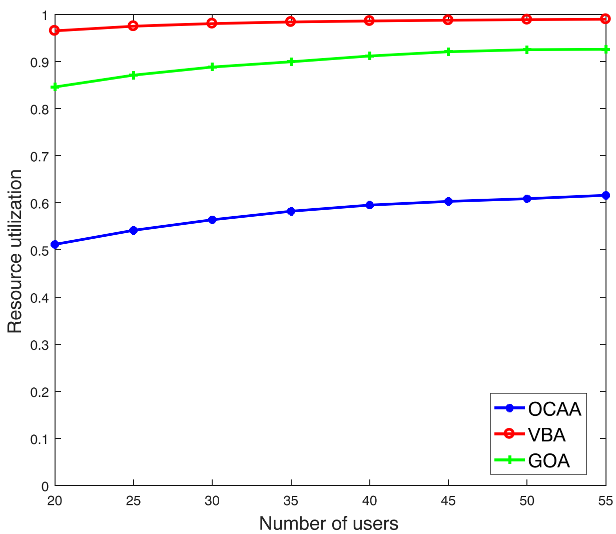

FIGURE 5 The resources utilization in a small-capacity network

about 50%. However, because of the coupling effect between computing and communication, it is not difficult to infer that no algorithm can reach 100% in resource utilization rate. For the algorithm OCAA, because of the existence of grouping, when request time of each user is uneven, the resource utilization rate is low. In addition, in order to determine the group payment, the user who has the lowest time bid in one group is eliminated, which results in a lower resource utilization for OCAA.

In Figure 6, the user payments of the proposed algorithm OCAA and VBA are compared for a small-capacity network scenario. Herein, OCAA-R represents the situation that only resource bidding payment is counted for the OCAA algorithm, while OCAA-RT counts both the time bidding payment and resource bidding payment for OCAA . In addition, the payment rule of VBA is from the VCG mechanism. [20] Since GOA cannot obtain the optimal solution of the problem maximizing social welfare, hence its user payment is not discussed here. From Figure 6, we note that as the number of users increases, the total user payment becomes larger. The reason is that the larger the number of users is, the easier the resources is allocated to high-value users, thus it comes to a larger social welfare and get higher revenues. In addition, we note that the difference of user payments between OCAA and VBA is not so large when only resource payment is counted for the OCAA. However, if both the resource and time priority bidding payments are considered for the OCAA, in order to obtain the resources, the users will pay more, which leads to higher revenues for the service providers.

The social welfare and resources utilization of the proposed three algorithms are also analyzed for the large-capacity network scenario, and the results are shown in Figures 7and 8, respectively. Different from the small-capacity network,

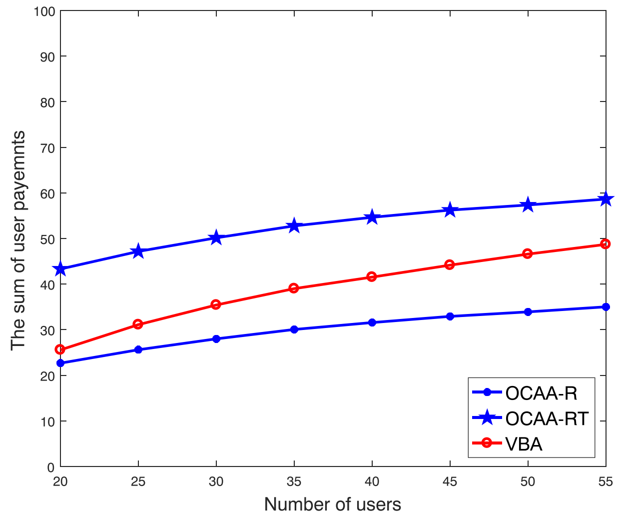

FIGURE 6 The sum of user payments in a small-capacity network

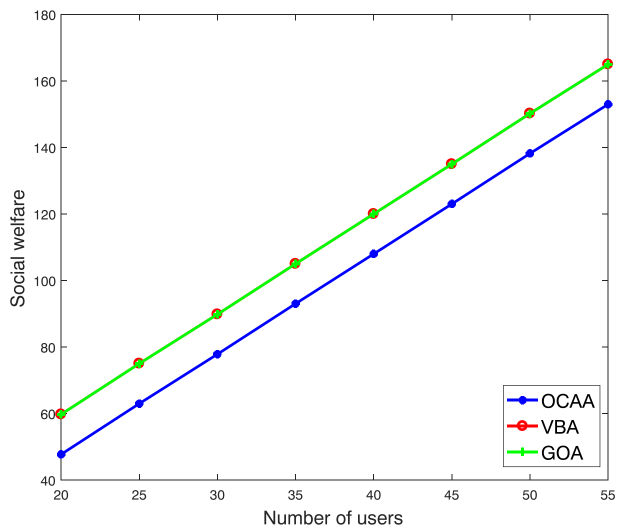

FIGURE 7 The social welfare in a large-capacity network

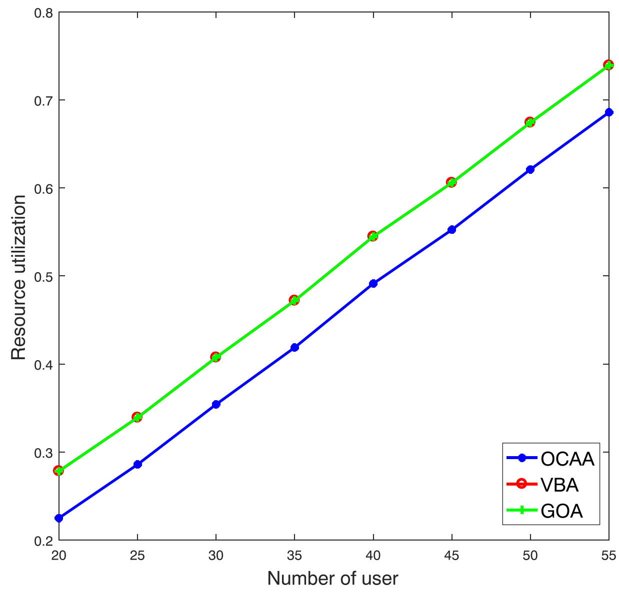

FIGURE 8 The resources utilization in a large-capacity network

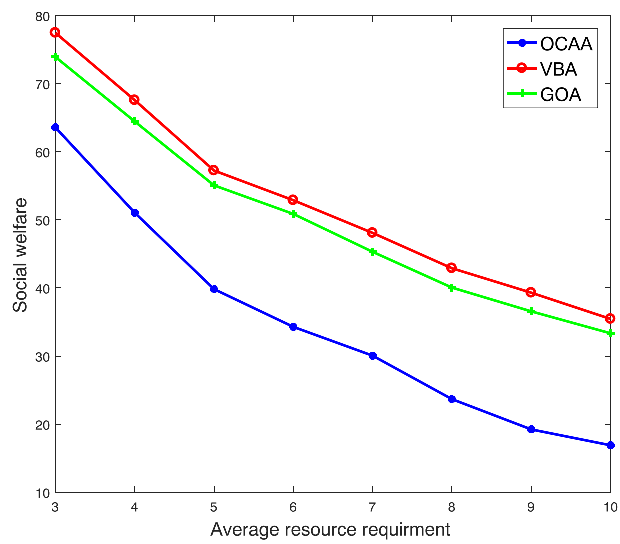

FIGURE 9 The social welfare versus the average requested resource

the social welfare and resources utilization of the three algorithms increase linearly with the increase of the number of users. This phenomenon can be explained as follows. In large-capacity network, with the increasing of number of users, the number of users who successfully allocated resources is increased. As a result, social welfare and resources utilization ratios present a nearly linear growth trend. In addition, in the large-capacity network, because of the sufficient resources, the competition between users is weak. Furthermore, the difference of social welfare and resource utilization between OCAA and GOA (or VBA) are small.

In Figure 9, the social welfare is evaluated by varying the average amount of resource asked by each user (including computing and communication resource). Now, the network computing capacity and communication capacity are both 25, the number of users is 40, the average computing resource and communication resources requested by each user is in the range [3, 9], and other parameters are the same as above. We can observe that if more resource is requested by the user, the social welfare of all these three algorithms are decreased. The reason is simple as that the increase in the user resources demands leads to a decrease in the number of users that can be served by the system. However, any other changes of the social welfare for these three algorithms are consistent with Figure 4.

In Figure 10, the sum of user payments are analyzed by varying the average amount of resource asked by the users. From Figure 10, we note that, as expected, the total payment of the OCAA decreases when users request more resources.

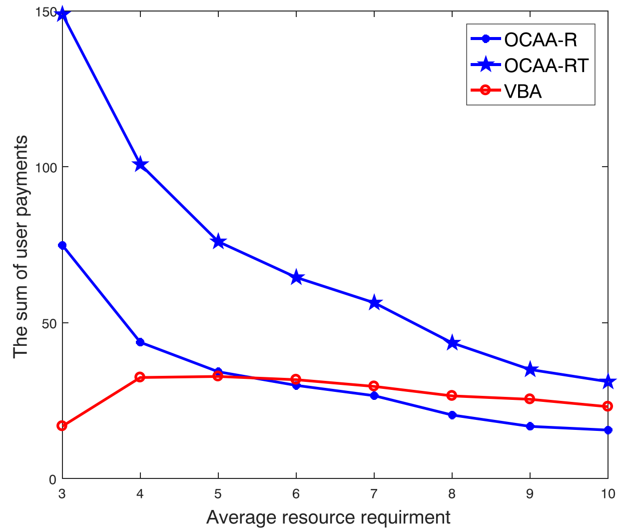

FIGURE 10 The sum of user payments versus the average requested resource

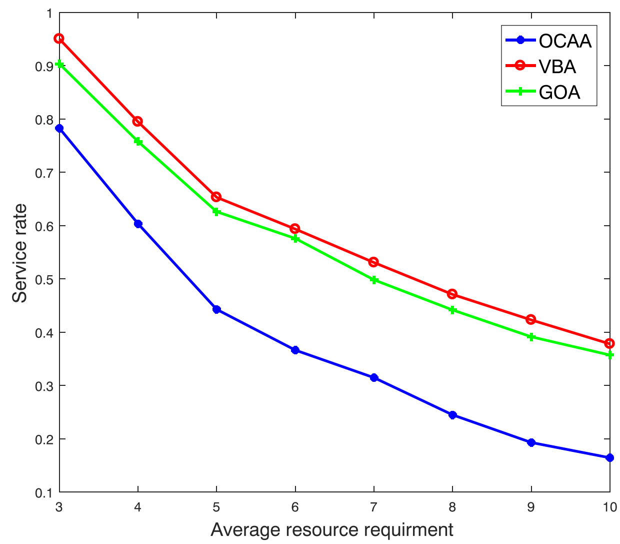

FIGURE 11 The service rate versus the average requested resource

This phenomenon can be explained as that with the increasing of the amount of requested resource, the number of users that can be allocated resource are reduced, thus the total payment of the OCAA is decreased. However, it is surprising that the total payment of the VBA algorithm shows a trend of increasing first and then decreased. In fact, this phenomenon is due to the fact that the VCG mechanism-based payment rules are adopted by VBA. In essence, the payment of the VCG mechanism is the loss of the remaining users. Therefore, when the average requested resource is less than 4, the system resources is sufficient enough, the competition among users is weak, and the potential loss caused by the participation of a certain user to the rest of the users is small, thus the payment is low and the trend is increasing. While when the average requested resources is greater than 4, the network resources is insufficient, and the number of users who have been allocated resources is greatly reduced, which results in a decrease in total payment. In contrast, compared with the VBA, OCAA has a higher total user payment because of the additional time bidding.

At last, the users' service rate (the number of users that have been allocated resources divided by the total number of users) is analyzed by varying the average amount of requested resource for the users, and the result is shown in Figure 11, from which, we observe that, the users' service rate of all the three algorithms are decreased as the amount of resources requested by the user increase. We can explain this phenomenon as follows. The increase in users' resources demands leads to a decrease in the number of users that the system can serve. In addition, the users' service rate of the OCAA algorithm is lower than GOA and VBA; in order to ensure high revenue and better economical properties, the grouping process in OCAA will reject some users, which leads to a lower service rate.

Through the analysis of the above simulation results, we can know that OCAA algorithm can bring high revenue to the service providers. However, the social welfare and resource utilization of OCAA algorithm is lower than the optimization-based algorithm GOA and VBA. Therefore, the operator can choose the suitable algorithm for their networks.

6 | CONCLUSION

In this paper, by jointly considering the time-varying characteristics and the combinatorial effect of the resource demands for the users in the MEC networks, the online combinatorial resources allocation problem is discussed herein. A fixed-time segmentation-based online resources allocation strategy is presented, and then a combinatorial auction-based mechanism is proposed for the computing and communication resources allocation in the fixed-time period. We prove that the proposed auction mechanism satisfies both the individual rational and incentive compatibilities, and the proposed mechanism can bring a high revenue to the service provider. In order to evaluate the performance of our proposed auction-based algorithm, a performance-bound algorithm based on variable relaxation and a suboptimal algorithm based on greedy thought are proposed. The simulation results demonstrate that the online combinatorial auction algorithm can bring higher revenues to service providers, while the optimization-based algorithms bring higher social welfare and resource utilization.

DATA AVAILABILITY (EXCLUDING REVIEW ARTICLES)

The studies and conclusions of this paper are supported by MATLAB simulation results and are included within the article.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

FUNDING INFORMATION

This work was supported in part by the Core Electronic Devices, High-end General Chips and Basic Software Products Projects of China under Grant 2017ZX01030204, in part by Sichuan Science and Technology Program under Grant 2018GZ0228, in part by Chongqing Frontier and Applied Basic Research Project under Grant cstc2015jcyjA40021 and cstc2017jcyjBX0047, and in part by the Fundamental Research Funds for the Central Universities under Grant 106112016CDJXY500002.

ORCID

Weiheng Jiang https://orcid.org/0000-0002-1856-8337

REFERENCES

1. Mao Y, You C, Zhang J, Huang K, Letaief KB. A survey on mobile edge computing: the communication perspective. *IEEE Commun Surv Tutorials*. 2017;19(4):2322-2358.
2. Satyanarayanan M, Bahl V, Caceres R, Davies N. The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Comput*. 2009;8(4):14-23.
3. Patel M, Naughton B, Chan C, Sprecher N, Abeta S, Neal A. Mobile-edge computing introductory technical white paper. White Paper, Mobile-edge Computing (MEC) Industry Initiative; 2014:3-5.
4. Zhang H, Liu N, Chu X, Long K, Aghvami A-H, Leung VC. Network slicing based 5G and future mobile networks: mobility, resource management, and challenges. *IEEE Commun Mag*. 2017;55(8):138-145.

5. Chen M-H, Liang B, Dong M. A semidefinite relaxation approach to mobile cloud offloading with computing access point. In: Signal Processing Advances in Wireless Communications (SPAWC), 2015 IEEE 16th International Workshop on; 2015; Stockholm, Sweden:186-190.
6. Chen M-H, Liang B, Dong M. Joint offloading decision and resource allocation for multi-user multi-task mobile cloud. In: Communications (ICC), 2016 IEEE International Conference on; 2016; Kuala Lumpur, Malaysia:1-6.
7. Cheng J, Shi Y, Bai B, Chen W. Computation offloading in cloud-ran based mobile cloud computing system. In: Communications (ICC), 2016 IEEE International Conference on; 2016; Kuala Lumpur, Malaysia:1-6.
8. Yu Y, Zhang J, Letaief KB. Joint subcarrier and CPU time allocation for mobile edge computing. In: Global Communications Conference (GLOBECOM), 2016 IEEE; 2016; Washington, DC, USA:1-6.
9. Wang X, Wang J, Wang X, Chen X. Energy and delay tradeoff for application offloading in mobile cloud computing. *IEEE Syst J*. 2017;11(2):858-867.
10. Dinh TQ, Tang J, La QD, Quek TQ. Offloading in mobile edge computing: task allocation and computational frequency scaling. *IEEE Trans Commun*. 2017;65(8):3571-3584.
11. You C, Huang K, Chae H, Kim B-H. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Trans Wirel Commun*. 2017;16(3):1397-1411.
12. Wang Y, Sheng M, Wang X, Wang L, Li J. Mobile-edge computing: partial computation offloading using dynamic voltage scaling. *IEEE Trans Commun*. 2016;64(10):4268-4282.
13. Cao X, Wang F, Xu J, Zhang R, Cui S. Joint computation and communication cooperation for mobile edge computing, arXiv preprint arXiv:1704.06777; 2017.
14. Wang F, Xu J, Ding Z. Optimized multiuser computation offloading with multi-antenna NOMA. In: Globecom Workshops (GC WKSHPS), 2017 IEEE; 2017; Singapore, Singapore:1-7.
15. Sardellitti S, Scutari G, Barbarossa S. Joint optimization of radio and computational resources for multicell mobile-edge computing. *IEEE Trans Signal and Inf Process Over Netw*. 2015;1(2):89-103.
16. Al-Shuwaili A, Simeone O, Bagheri A, Scutari G. Joint uplink/downlink optimization for backhaul-limited mobile cloud computing with user scheduling. *IEEE Trans Signal Inf Process Over Netw*. 2017;3(4):787-802.
17. Chen X. Decentralized computation offloading game for mobile cloud computing. *IEEE Trans Parallel Distrib Syst*. 2015;26(4):974-983.
18. Chen X, Jiao L, Li W, Fu X. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Trans Networking*. 2016;24(5):2795-2808.
19. Zhang H, Qiu Y, Long K, Karagiannidis GK, Wang X, Nallanathan A. Resource allocation in NOMA based fog radio access networks, arXiv preprint arXiv:1803.05641; 2018.
20. Krishna V. *Auction Theory*: Academic press; 2009:1-8.
21. Wu F, Vaidya N. A strategy-proof radio spectrum auction mechanism in noncooperative wireless networks. *IEEE Trans Mob Comput*. 2013;12(5):885-894.
22. Chen Y, Lin P, Zhang Q. Lotus: Location-aware online truthful double auction for dynamic spectrum access. In: Dynamic Spectrum Access Networks (DYSPAN), 2014 IEEE International Symposium on; 2014; McLean, VA, USA:510-518.
23. Zhang H, Du J, Cheng J, Long Keping, Leung VC. Incomplete CSI based resource optimization in SWIPT enabled heterogeneous networks: a non-cooperative game theoretic approach. *IEEE Trans Wirel Commun*. 2018;17(3):1882-1892.
24. Zhang H, Guo F, Ji H, Zhu C. Combinational auction-based service provider selection in mobile edge computing networks. *IEEE Access*. 2017;5:13455-13464.
25. Xu K, Zhang Y, Shi X, Wang H, Wang Y, Shen M. Online combinatorial double auction for mobile cloud computing markets. In: Performance Computing and Communications Conference (IPCCC), 2014 IEEE International; 2014; Austin, TX, USA:1-8.
26. Mashayekhy L, Nejad MM, Grosu D, Vasilakos AV. An online mechanism for resource allocation and pricing in clouds. *IEEE Trans Comput*. 2016;65(4):1172-1184.
27. Xu L, Wang J, Nallanathan A, Li Y. Resource allocation based on double auction for cloud computing system. In: High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SMARTCITY/DSS), 2016 IEEE 18th International Conference on; 2016; Sydney, NSW, Australia:1538-1543.
28. Boyd S, Vandenberghe L. *Convex Optimization*: Cambridge university press; 2004:127-136.
29. Feng W-J, Jiang W. Primary network interference compensation-based dynamic spectrum leasing and secondary network power control. *Wirel Pers Commun*. 2015;82(2):965-992.
30. CVX. Matlab software for disciplined convex programming. http://cvxr.com/cvx/, Accessed April 4, 2018.
31. Feng W-J, Jiang W. Primary network interference compensation-based dynamic spectrum leasing and secondary network power control. *Wirel Pers Commun*. 2015;82(2):965-992.
32. Nisan N, Roughgarden T, Tardos E, Vazirani VV. *Algorithmic Game Theory*. Leiden: Cambridge University Press; 2007.
33. Jiang W, Gong Y, Cao Y, Wu X, Xiao Q. Energy-delay-cost tradeoff for task offloading in imbalanced edge cloud based computing, arXiv preprint arXiv:1805.02006; 2018.

How to cite this article: Wu X, Jiang W, Zhang Y, Yu W. Online combinatorial based mechanism for MEC network resource allocation. *Int J Commun Syst*. 2019;e3928. https://doi.org/10.1002/dac.3928