Computer Communications 181 (2022) 90–101

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Auction design for cross-edge task offloading in heterogeneous mobile edge clouds☆

Weifeng Lu [a], Weiduo Wu [a], Jia Xu [a,*], Pengcheng Zhao [a], Dejun Yang [b], Lijie Xu [a]

[a] Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, 210023, China
[b] Colorado School of Mines, Gordon, CO 80401, USA

ARTICLE INFO

Keywords:
Mobile edge computing
Task offloading
Incentive mechanism
Auction

ABSTRACT

Task offloading is a promising technology to exploit the available resources in edge cloud efficiently. Many incentive mechanisms for offloading systems have been proposed. However, most of existing works study the centralized incentive mechanisms under the assumption that all mobile edge infrastructures are operated by a central cloud. In this paper, we aim to design the auction-based truthful incentive mechanisms for heavily loaded task offloading system in heterogeneous MECs. We first study the homogeneous MEC situation and present a global auction executed in the central cloud as a benchmark. For the heterogeneous MEC situation, we model the system as a dual auction framework, which enables the heterogeneous MECs to perform cross-edge task offloading without the participation of central servers. Specifically, we design two dual auction models: secondary auction-based model, which enables the system to offload tasks from a large-scale region in a single auction, and double auction-based model, which is suitable for the time sensitive tasks. Then the auctions for these two dual auction models are proposed. Through rigorous theoretical analysis, we demonstrate that the proposed auctions achieve desirable properties of computational efficiency, individual rationality, budget balance, truthfulness, and guaranteed approximation. The simulation results show that the secondary auction and double auction can obtain 14.5% and 4.2% more social welfare than comparison algorithm on average, respectively. In addition, the double auction has great advantage in terms of computation efficiency.

1. Introduction

The pervasive proliferation of mobile devices enables mobile users to enjoy many interesting and useful applications. Due to the limited capacity of mobile devices and the increasing resource requirement of applications, the computation and storage resources of mobile devices become insufficient. With the large proliferation of time sensitive services such as industrial IoT [1], real-time video/image processing [2], and AR/VR applications [3], mobile edge computing has emerged to provide the computation and storage resources at the edge of network [4].

Generally, the tasks are with different resource requirements, such as CPU frequency, bandwidth, and storage. Task offloading to the Mobile Edge Clouds (MEC) is an efficient way to improve the performance of system and reduce the resource consumption of mobile devices [5]. A lot of efforts [6,7] have focused on developing multi-user task offloading system in MEC. Incentive mechanism is important in most human involved systems [8–11]. For the task offloading system, incentive mechanism is crucial as the rational MECs will not provide the resources unless they can obtain sufficient compensation. Some recent efforts [12–14] have focused on developing incentive mechanisms to encourage edge cloud to participate in task offloading.

However, most of existing studies [15,16] consider that all MECs are operated by the central cloud, which manages MECs as a central server, and develop the centralized incentive mechanisms for task offloading. In this paper, the MECs, which are operated by a single service provider are termed homogeneous MECs. In real life, the MECs are operated by different service providers. For example, the base stations and APs are operated by different telecom companies, such as AT&T, Verizon, Sprint and T-Mobile, even these edge infrastructures are close to each other. The MECs are heterogeneous since they are operated by multiple service providers. The centralized resource scheduling algorithms and incentive mechanisms will not be effective in this heterogeneous MEC scenario. There are two special issues in heterogeneous MEC task offloading system:

☆ This research was supported in part by Natural Science Foundation of China grants 61872193, 61872191 and 62072254, and Natural Science Foundation of the United States grants 1717315.
* Corresponding author.
E-mail addresses: luwf@njupt.edu.cn (W. Lu), 1218043211@njupt.edu.cn (W. Wu), xujia@njupt.edu.cn (J. Xu), 2019070272@njupt.edu.cn (P. Zhao), djyang@mines.edu (D. Yang), ljxu@njupt.edu.cn (L. Xu).

https://doi.org/10.1016/j.comcom.2021.09.035
Received 26 February 2021; Received in revised form 9 July 2021; Accepted 28 September 2021
Available online 6 October 2021
0140-3664/

The first problem is the load migration among heterogeneous MECs. With the development of emerging smart grid, intelligent transportation, industrial IoT, etc., task offloading is becoming more and more pervasive. Load migration is indispensable for the heavily loaded task offloading system. However, there is no central server in heterogeneous MEC scenario, and the centralized load balancing mechanisms [17,18] are invalid. The new load migration mechanism working on the edge side is essential.

The second problem is the strategic behavior of heterogeneous MECs. Since the MECs are operated by different service providers, who usually compete with each other, the MECs may take strategic behavior by submitting dishonest cost to maximize their utilities.

In this paper, we aim to design auction-based truthful incentive mechanisms for heavily loaded task offloading system. We first study the homogeneous MEC situation and present a global auction executed in the central cloud as the benchmark of the auctions proposed in heterogeneous MEC situation. For the heterogeneous MEC situation, we model the system as a dual auction, which can achieve cross-edge task offloading without the participation of central servers, to transfer the task load to heterogeneous MECs through the profit-driven cooperation. We further design two dual auction models for the heterogeneous MEC scenario: secondary auction-based model and double auction-based model. In secondary auction model, the idle MEC launches an auction to sell its idle resources to the mobile users who failed in the local task offloading. In double auction-based model, the busy MEC launches a double auction to facilitate the transactions between the idle MECs and mobile users.

The main contributions of this paper are as follows:

- We design the truthful incentive mechanisms for cross-edge task offloading to stimulate the heterogeneous and strategic MECs.
- We design the system model for task offloading in homogeneous MECs, and formulate the social optimal task offloading problem to maximize the social welfare. We present a global auction as the benchmark of the auctions proposed in heterogeneous MEC situation.
- For the heterogeneous MECs, we model the task offloading system as a dual auction, enabling the heterogeneous MECs to perform cross-edge task offloading without the participation of central servers. We present two dual auction models: secondary auction-based model and double auction-based model. The secondary auction-based model enables the system to offload tasks of mobile users from a large-scale region in a single auction. The double auction-based model can offload the tasks in time and is suitable for the time sensitive tasks.
- We present the local auction, secondary auction, and double auction for the task offloading in heterogeneous MECs. We show that textcolorredthe local auction and secondary auction satisfy the desirable properties of computational efficiency, individual rationality, truthfulness, and approximation ratio. The double auction can satisfy the properties of computational efficiency, individual rationality, budget balance, and truthfulness.

The rest of the paper is organized as follows. We review the state-of-art research in Section 2. Section 3 formulates the system models and problems, and lists some desirable properties. Section 4, Section 5, and Section 6 present the detailed design and analysis of our auctions for homogeneous MEC model, secondary auction-based heterogeneous MEC model and double auction-based heterogeneous MEC Model, respectively. Performance evaluation is presented in Section 7. We conclude this paper in Section 8.

2. Related work

2.1. Resource scheduling in task offloading

The concept of task offloading was proposed as a promising solution for emerging computation-intensive and real time services. Tran et al. consider the MEC enabled multi-cell wireless network and formulate the problem of joint task offloading and resource allocation as a mixed integer non-linear program under the constraints of computing resource and uplink transmission power [5]. They decompose the original problem into a resource allocation problem with fixed task offloading decision and a task offloading problem that optimizes the optimal-value function corresponding to the resource allocation problem. Chen et al. investigate the task offloading problem in ultra-dense network and aim to minimize the delay while saving the battery life of user's equipment [16]. They formulate the task offloading problem as a mixed integer non-linear program, and also transform the program into two sub-problems (task placement sub-problem and resource allocation sub-problem). Rahman et al. consider the task offloading in cloud networked multi-robot systems [19]. They take the delay constraint, the extra costs of data transmission and remote computation into consideration. Then, an integrated framework, which owes a significant improvement in overall system performance for cloud networked multi-robot systems, is proposed. Guo et al. consider the energy-efficient computation offloading under hard constraint of application completion time [20].

However, the aforementioned works do not consider the load migration among the MECs. Since both the load and the available resources are time varying in practice, cross-edge load migration is necessary in task offloading systems.

2.2. Load balancing in task offloading

The load balancing mechanism aims at improving the overall workload of MEC system. Li et al. divide the MECs into three categories (light-load MEC, normal-load MEC, and heavy-load MEC) and study the balancing among different MECs to reduce the execution time of tasks [18]. Laredo et al. investigate a self-organized critical approach for dynamically load-balancing computational workload [21]. The proposed model shows the features of load balancing and improves the resource utilization. Zeng et al. try to minimize the total operating cost in caching popular contents using load balancing [17]. They design an online algorithm to solve the joint problem of content placement and load balancing.

However, the works above mentioned do not consider the incentive mechanism for the MECs. Since the MECs are operated by different service providers, the MECs may take strategic behavior by submitting dishonest cost to maximize their utilities.

2.3. Incentive mechanism for task offloading

Most of existing studies of incentive mechanism in MEC aim at stimulating all parties to participant in the task offloading system. Liu et al. consider the incentive mechanism between edge clouds and central cloud [9]. They formulate the interactions among central cloud and edge clouds as a Stackelberg game to maximize the utilities of central cloud and edge clouds by obtaining the optimal payment and computation offloading strategies. Li et al. propose an online truthful mechanism for computation and communication resource allocation [12]. In their system model, upon the arrival of a smartphone user who requests task offloading, the base station needs to make a decision right away without knowing any future information.

Auction is an efficient market mechanism to allocate the task and determine the price and is widely used in crowdsensing, cloud computing, and so on. Xu et al. focus on exploring truthful incentive mechanisms for a novel and practical scenario of crowdsensing, where the tasks are time window dependent, and the platform has strong requirement of data integrity [22,23]. They formulate the problem as the social optimization user selection problem and design two incentive mechanisms. Tan et al. study a general online combinatorial auction problem, in which a provider allocates multiple types of capacity-limited resources to customers who arrive in a sequential and arbitrary manner [24]. Samimi et al. propose the combinatorial double auction resource allocation, which solves the problem that most of the current market-based resource allocation models are biased in favor of the provider over the buyer in an unregulated trading environment [25].

Recently, auction has been used for task offloading in edge computing. Habiba et al. propose a reverse auction framework based on position auction, consisting of pricing, bidding strategy optimization, and winner determination [26]. The auction aims to maximize the utility of MEC through strategic participation and can obtain the desirable economical characteristics of envy-free and individual rationality. Zhang et al. propose an auction scheme for computation resource allocation in D2D-assisted MEC system [27].

However, aforementioned works study the centralized incentive mechanisms under the assumption that all mobile edge infrastructures are operated by the central cloud.

Overall, there is no off-the-shelf research in the literature, which study the decentralized incentive mechanism for cross-edge task offloading in heterogeneous and strategic MECs.

3. System model and desirable properties

In this section, we consider two kinds of models for heavily loaded task offloading systems: homogeneous MEC model and heterogeneous MEC model. In homogeneous MEC model, all MECs are operated by the same service provider, and the resources of homogeneous MECs are scheduled by the uniform central cloud. In heterogeneous MEC model, the MECs are operated by different service providers, i.e., there is no uniform central cloud to schedule the resources for task offloading. Different from most existing task offloading system, we model the task offloading in heterogeneous MECs as a dual auction, allowing the cooperative task offloading between the MECs. We consider that each MEC is selfish to other MECs in heterogeneous MEC model when bidding for an auction. Moreover, the required resources are always more than the available resources provided in the designed auctions since we consider the heavily loaded task offloading.

3.1. Homogeneous MEC model

We consider a task offloading system consisting of a central cloud, a set of m MECs $E = \{e_1, e_2, \dots, e_m\}$, and a set of n mobile device users $U = \{u_1, u_2, \dots, u_n\}$. Let $m \ll n$ since we consider the heavily loaded task offloading system.

Each user $u_j \in U$ has a task t_j, which is expected to be offloaded to the MEC. We consider that the task is indivisible, and can only be offloaded to at most one MEC. We denote the set of tasks as $T = \{t_1, t_2, \dots, t_n\}$. We consider that there are r kinds of resources in the MECs, denoted as $R = \{1, 2, \dots, r\}$, such as internal storage, external storage, CPU frequency, bandwidth, and GPU memory. Each task t_j has a resource requirement $\boldsymbol{a_j}= (a_j^1, a_j^2, \dots, a_j^r)$, where a_j^i is the requirement of any resource $i \in R$. Specifically, if there is no requirement of some resource, the corresponding requirement could be set as zero. Note that the requirement of each kind of resources is normalized by the benchmark unit. The benchmark units of every kind of resources have the same value. For example, if the value of 1GB internal storage equals to the value of 100GB external storage, we can set 1GB and 100GB as the benchmark units of internal storage and external storage, respectively. Each MEC $e_k \in E$ has a resource capacity $\boldsymbol{d_k}= (d_k^1, d_k^2, \dots, d_k^r)$, which is a vector of available resources. Similarly, the resource capacity is also normalized by the benchmark unit.

We model the task offloading system as a sealed auction. First, each user $u_j \in U$ submits a bid $B_j = (t_j,\ \boldsymbol{a_j},\ b_j)$ to one of available MECs based on its location, preference and so on. b_i is user u_j's bid price, i.e., the maximal price it can pay for offloading task t_j. Each user u_j also has a value v_j for offloading task t_j. We consider that v_j is the private information and is known only to user u_j. Actually, the user may have multiple tasks to be offloaded at same time. In this case, the user can be viewed as multiple users, and each user bids for one task independently.

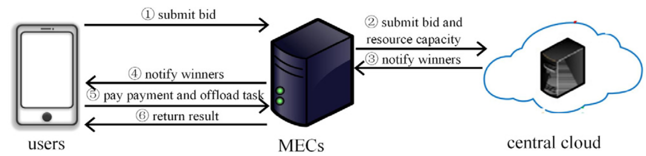


Fig. 1. Task offloading of homogeneous MEC model.

We consider that the mobile users cannot communication with the central cloud directly. After receiving the bids from users, each MEC submits the bids and the resource capacity of itself to the central cloud. We consider that the MECs are honest but rational to the users, i.e., the MECs never tamper with bids of users. Given the user set U, the bid profile $\mathbf{B}= (B_1, B_2, \dots, B_n)$, and the resource capacity matrix $\mathbf{D} = (\boldsymbol{d_1}, \boldsymbol{d_2}, \dots, \mathbf{d_m})$, the central cloud calculates the winning set $S \in U$, the resource allocation profile $\mathbf{q}= (q_1, q_2, \dots, q_n)$, and the payment profile $\mathbf{p}= (p_1, p_2, \dots, p_n)$, where q_j and p_j, $j = 1, 2, \dots, n$, are the allocated MEC and the payment of user u_j, respectively. Then the central cloud notifies winners of the determination via MECs. Each winner u_j pays p_j to MEC q_j, and offloads task t_j to q_j. Finally, q_j returns the result to u_j when it completes t_j. The whole process is illustrated by Fig. 1.

We define the utility of user u_j as the difference between the value and payment:

$$ut_j = \begin{cases} v_j - p_j, & \text{if } u_j \in S \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Specially, the utility of the losers would be zero because they pay nothing and there is no value for them.

Note that b_j can be different from the value v_j because we consider the users selfish. So, the users may take a strategic behavior by claiming dishonest value to maximize their own utilities.

Sicne all MECs are operated by the same service provider in homogeneous MEC model, we define the utility of the central cloud as:

$$ut_0 = \sum_{u_j \in S} p_j \tag{2}$$

Moreover, the social welfare is:

$$ut_s = ut_0 + \sum_{u_j \in S} ut_j = \sum_{u_j \in S} v_j \tag{3}$$

The objective is maximizing the social welfare subject to the constraint that each task is offloaded to at most one MEC, and all the resource requirements of winners can be satisfied. Although the value v_j is only known by user u_j, we will prove that claiming a different value b_j cannot help to increase the utility of user u_j in our designed mechanisms. So, we still use b_j when we attempt to maximize the social welfare. We call this problem as *Global Social Optimal Task Offloading (GSOTO)* problem, which can be formulated as follows:

$$\max \sum_{e_k \in E} \sum_{u_j \in U} b_j y_k^j \tag{4}$$

$$s.t. \quad \sum_{u_j \in U} a_j^i y_k^j \le d_k^i, \forall i \in R, \forall e_k \in E \tag{4-1}$$

$$\sum_{e_k \in E} y_k^j \le 1, \forall u_j \in U \tag{4-2}$$

$$y_k^i \in \{0, 1\}, \forall u_j \in U, \forall e_k \in E \tag{4-3}$$

where $y_k^j = 1$ represents that task t_j is offloaded to MEC e_k, and $y_k^j = 0$ otherwise. The constraint (4–1) ensures that the total resource requirement is no more than the resource capacity for every resource in each MEC. The constraint (4–2) ensures that a task can only be offloaded to at most one MEC, i.e., the task is indivisible.

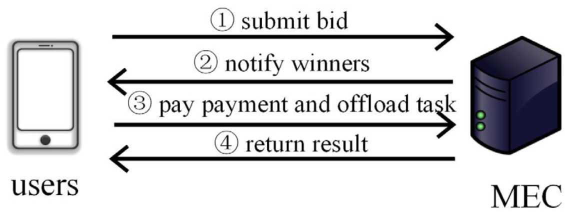


Fig. 2. Process of local auction.

3.2. Secondary auction-based heterogeneous MEC model

In this subsection, we consider that the MECs are operated by different service providers. This means that there is no central cloud to hold the auction. To address this problem, we model the task offloading in heterogeneous MECs as a dual auction system consisting of local auction and secondary auction. The secondary auction-based heterogeneous MEC model enables the system to offload tasks of mobile users from a large-scale region in a single auction. Thus, the secondary auction-based heterogeneous MEC model is suitable for the large-scale task offloading system.

First, the users submit their bids for task offloading to the corresponding MECs. Different from the homogeneous MEC model, there is no central controller in the heterogeneous MEC model. Thus, the users cannot communicate with the MECs in other areas. However, the MECs can communicate with each other through backbone network. For example, the base stations are interconnected by the backbone network, even they are operated by different service providers. Since the MECs are operated by different service providers, the MECs always do not share the revenue with other MECs. At the beginning of task offloading, these heterogeneous MECs have conflicts of interest, and they all want to obtain the tasks to get the payment from the users. Thus, they will never offload the local tasks to other MECs, and cannot be controlled by any central controller.

Since there is no central controller, each MEC executes a local auction independently. We denote the user set and the corresponding bid profile of any MEC e_k as U^k and $\mathbf{B^k}$, respectively. Each MEC e_k calculates the winner set $S^k \subseteq U^k$ and the payment profile $\mathbf{p^k}$. Then Each MEC e_k notifies winners of the determination. Each winner $u_j \in S^k$ pays p_j to e_k, and offloads task t_j to e_k. Finally, e_k return the result to u_j when it completes t_j. The process of local auction is illustrated by Fig. 2.

We define the utility of any MEC e_k as:

$$ut_{e_k} = \sum_{u_j \in S^k} p_j \tag{5}$$

Then the social welfare is:

$$ut_s = ut_{e_k} + \sum_{u_j \in S^k} ut_j = \sum_{u_j \in S^k} v_j \tag{6}$$

The objective of local auction is maximizing the social welfare such that all the resource requirements of winners can be satisfied. We call this problem as *Local Social Optimal Task Offloading (LSOTO)* problem, which can be formulated as follows for each MEC e_k:

$$\max \sum_{u_j \in U} b_j y_k^j \tag{7}$$

$$s.t. \quad \sum_{u_j \in U^k} a_j^i y_k^j \le d_k^i, \forall i \in R \tag{7-1}$$

$$y_k^j \in \{0,1\}, \forall u_j \in U, \forall e_k \in E \tag{7-2}$$

After local auction, the MECs can be divided into two categories: busy MECs who still have tasks needed to be offloaded and idle MECs that are with residual resources. In order to offload tasks as many as possible and take full advantage of residual resources of idle MECs, the idle MECs can launch a secondary auction for the losers of local auction in busy MECs.

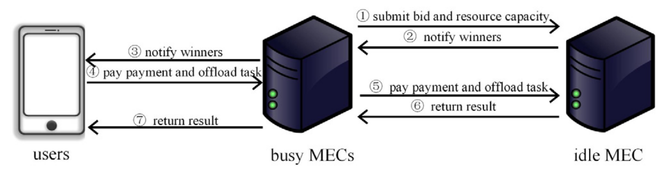


Fig. 3. Process of secondary auction.

Any idle MEC can launch a secondary auction and inform other MECs of the auction at any time. When a busy MEC receives the information, it submits the bids of losers in local auction to the idle MEC. The secondary auction is an idle MEC-led auction, which is fit for the heavily loaded task offloading system.

Basically, the process of secondary auction is similar to the local auction. The difference is that each busy MEC will take the agency fee $\alpha_k p_j$ for any task t_j to compensate the agency cost such as communication cost and store-and-forward cost, where $\alpha_k \in (0,1)$ is the agency fee rate of any busy MEC e_k. We consider that α_k is the public information. When receiving the payment from each winner of secondary auction, the corresponding busy MEC will deduct the agency fee, and submit the residual payment to the idle MEC. Note that we consider that the MEC is honest but rational to the users, i.e., the busy MECs never tamper with bids of users and never deduct more than the declared agency fee. The process of secondary auction is illustrated by Fig. 3.

Given the winner set $\overline{S^k}$ of busy MEC e_k in the secondary auction, the utility of each busy MEC e_k is:

$$ut_{e_k} = \sum_{u_j \in \overline{S^k}} \alpha_k p_j \tag{8}$$

We define the utility of the idle MEC as:

$$ut_{idle} = \sum_{e_k \in E} \sum_{u_j \in \overline{S^k}} (1-\alpha_k) p_j \tag{9}$$

Then the social welfare is:

$$ut_s = ut_{idle} + \sum_{e_k \in E} ut_{e_k} + \sum_{e_k \in E} \sum_{u_j \in \overline{S^k}} ut_j = \sum_{u_j \in \overline{S^k}} v_j \tag{10}$$

The objective of secondary auction is maximizing the social welfare subject to the resource constraint, which is same with local auction. Obviously, this problem is also the *LSOTO* problem, which has been formulated in (7).

3.3. Double auction-based heterogeneous MEC model

In this subsection, we present a cooperative task offloading model through double auction. Different from the secondary auction, the double auction is launched by a busy MEC once there are tasks, which cannot be offloaded in the local MEC. Thus this model can improve the timeliness, and is suitable for the time sensitive tasks. Moreover, as the sellers of double auction, the idle MECs are always profitable. We model the task offloading system as a dual auction system consisting of local auction and double auction.

The local auction is same with that of secondary auction-based heterogeneous MEC model. Then we consider that a busy MEC e_0 launches a double auction, where the idle MECs are sellers, and the users in $U^L = U^0 \backslash S^0$, i.e., the losers of local auction in MEC e_0, are the buyers. After that, each idle MEC e_k in the idle MEC set E^{idle} submits an ask $A_k = (\boldsymbol{d_k}, g_k)$ to e_0, where g_k is the unit resource ask price of e_k.

Given the user set U^L, the idle MEC set E^{idle}, the bid profile $\mathbf{B^L}$ of users in U^L, and the ask profile $\mathbf{A^{idle}}$ of idle MECs in E^{idle}, the busy MEC calculates the winning user set $S^U \subseteq U^L$ the winning idle MEC set $S^E \subseteq E^{idle}$, the resource allocation profile $\mathbf{q}$, and the payment profile

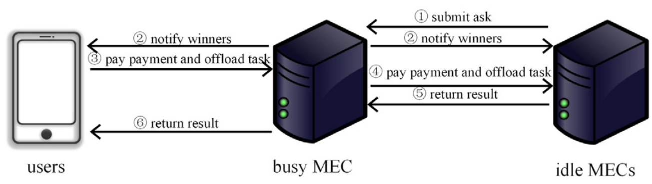


Fig. 4. Process of double auction.

$\mathbf{p} = (\mathbf{p^U}, \mathbf{p^E})$, where $\mathbf{p^U} = (p_1^U, p_2^U, \dots, p_{|U^L|}^U)$ and $\mathbf{p^U} = (p_1^E, p_2^E, \dots, p_{|E^{idle}|}^E)$ are the payment of users and idle MECs, respectively. Then the busy MEC notifies winners of the determination. Each winner $u_j \in S^U$ pays p_j^U to busy MEC e_0, and e_0 pays p_k^E to each winning idle $e_k \in S^E$. Afterwards, each winner u_j offloads task t_j to MEC q_j via busy MEC. Finally, q_j return the results to u_j via busy MEC when it completes t_j. The whole process is illustrated by Fig. 4.

We define the utility of user u_j as the difference between the value and payment:

$$ut_j = \begin{cases} v_j - p_j^U, & \text{if } u_j \in S^U \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

The utility of each idle MEC e_k is:

$$ut_{e_k} = \begin{cases} p_j^U - c_k \sum_{u_j \in S^U,\ q_j = e_k} \sum_{i \in R} a_j^i, & \text{if } e_k \in S^E \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where c_k is the unit resource cost of e_k.

Note that b_j can be different from the value v_j and g_k can be different from the cost c_k because we consider both users and idle MECs selfish. So, they may take a strategic behavior by claiming dishonest value/cost to maximize their own utilities.

The utility of busy MEC e_0 is:

$$ut_{e_0} = \sum_{u_j \in S^U} p_j^U - \sum_{e_k \in S^E} p_k^E \tag{13}$$

3.4. Desirable properties

Our objective is to design the auctions satisfying the following desirable properties:

- **Computational Efficiency:** An auction is computationally efficient if the winner set S, the resource allocation $\mathbf{q}$, and the payment $\mathbf{p}$ can be computed in polynomial time.
- **Individual Rationality:** Each bidder will have a non-negative utility while reporting true private information.
- **Truthfulness:** An auction is truthful if no bidder can improve its utility by submitting false value/cost, no matter what others submit. In other words, reporting the real value/cost is a weekly dominant strategy for all bidders.
- **Approximation Ratio:** The goal of the auction is to maximize the social welfare. We attempt to find the algorithms with low approximation ratios.

The importance of the first three properties is obvious, because they together ensure the feasibility of the auction. The last two properties are indispensable for guaranteeing the compatibility and high performance. Being truthful, the auction can eliminate the fear of market manipulation and the overhead of strategizing over others for the participating users.

We list the frequently used notations in Table 1.

4. Task offloading with homogeneous MECs

In this section, we present the global auction for the task offloading with homogeneous MECs as the benchmark. Furthermore, the global

Table 1
Frequently used notations.

Notations	Description
E	set of MECs
U, S	user set, winner set
T, t_j	task set, task of user u_j
$\boldsymbol{a}_j, a_j^i$	task t_j's resource requirement, task t_j's requirement of resource i
$\boldsymbol{d}_k, d_k^i$	MEC e_k's resource capacity, MEC e_k's capacity of resource i
v_j, b_j	value of user u_j, bid price of user u_j
$\mathbf{B}$, B_j	bid profile of users, bid of user u_j
$\mathbf{D}$	MECs' resource capacity profile
$\mathbf{q}$, q_j	resource allocation profile, allocated MEC of user u_j
$\mathbf{p}$, p_j	payment profile, payment of user u_j
ut_0, ut_j	utility of central cloud, utility of user u_j
ut_s	social welfare
U^k, S^k	set of user in MEC e_k, set of winners in MEC e_k
$\mathbf{B^k}$	bid profile of users in MEC e_k
$\mathbf{p^k}$	payment profile in MEC e_k
ut_{e_k}	utility of MEC e_k
α_k	agency fee of MEC e_k
U^L	loser set of local auction in busy MEC
$\mathbf{B^L}$	bid profile of loser in busy MEC
E^{idle}	set of idle MECs
g_k	unit resource ask price of idle MEC e_k
c_k	unit resource cost of idle MEC e_k
A_k	aks of idle MEC e_k
$\mathbf{A^{idle}}$	ask profile of all idle MECs
$\mathbf{p^U}, \mathbf{p^E}$	payment profile of losers, payment profile of idle MECs

auction presented in this section will be employed by the double auction for heterogeneous MECs.

4.1. Auction design

First of all, we attempt to find an efficient algorithm for the *GSOTO* problem presented in (4). Unfortunately, as the following theorem shows, it is NP-hard to find the optimal solution.

Theorem 1. *The GSOTO problem is NP-hard.*

Proof. We consider a corresponding instance of *GSOTO*: Let $T = \{t_1, t_2, \dots, t_n\}$ denote the set of tasks. Each task $t_j \in T$ has a resource requirement $\boldsymbol{a_j} = (a_j^1, a_j^2, \dots, a_j^r)$ and a value v_j. Let $E = \{e_1, e_1, \dots, e_m\}$ denote the set of MECs. Each MEC is with resource capacity $\boldsymbol{d_k} = (d_k^1, d_k^2, \dots, d_k^r)$. The target is to maximize the sum of the values of the tasks in the MECs so that the sum of resource requirement of tasks in any MEC $e_k \in E$ for every resource $i \in \{1, 2, \dots, r\}$ does not exceed the resource capacity d_k^i of MEC e_k. Then we can see that this problem is the *Multiple Multi-Dimension Knapsacks (MMDK)* problem [28]. Since the *MMDK* problem is a well-known NP-hard problem, the *GSOTO* problem is NP-hard. ■

Since the *GSOTO* problem is NP-hard, it is impossible to compute the winner set with maximal social welfare in polynomial time unless P=NP. In addition, we cannot use the off-the-shelf VCG mechanism [29] since the truthfulness of VCG mechanism requires that the social welfare is exactly maximized. Our global auction follows a greedy approach. As illustrated in Algorithm 1, global auction consists of winner selection phase and payment determination phase.

In the winner selection phase, the users are sorted according to the unit resource value, which is defined as $\frac{b_j}{\sum_{i \in R} a_j^i}$ for any user $u_j \in U$, in nonincreasing order (Line 2). The MECs are sorted according to the total resource capacity $\sum_{i \in R} d_k^i$ for any MEC $e_k \in E$ in nondecreasing order (Line 3). We iterate all users in order, and try to allocate the user to the MEC with minimum resource capacity in each iteration. If the resource capacity of the MEC can satisfy the resource requirement of the user, the user is selected as a winner, and the corresponding task is allocated to the MEC (Line 7); otherwise, we try to allocate the user to the next MEC in the order.

Algorithm 1: Global Auction

Input: user set U, bid profile $\mathbf{B}$ and resource capacity matrix $\mathbf{D}$
Output: winner set S, allocation profile $\mathbf{q}$, payment profile $\mathbf{p}$
//Phase 1: Winner Selection
1: $S \leftarrow \emptyset$; $\mathbf{D}' \leftarrow \mathbf{D}$; $\mathbf{q} \leftarrow 0$;
2: sort all users based on $\frac{b_j}{\sum_{i\in R} a_j^i}$ for $\forall j \in U$ in nonincreasing order and the sequence is denoted by Q_U;
3: sort all MECs based on $\sum_{i\in R} d_k^i$ for $\forall e_k \in E$ in nondecreasing order and the sequence is denoted by Q_E;
4: **foreach** $u_j \in Q_U$ in order **do**
5: **foreach** $e_k \in Q_E$ in order **do**
6: **if** $d'^i_k \geq a_j^i, \forall i \in R$ **then**
7: $S \leftarrow S \cup \{u_j\}$; $q_j \leftarrow e_k$;
8: **end if**
9: **foreach** $i \in R$ **do**
10: $d'^i_k \leftarrow d'^i_k - a_j^i$;
11: **end for**
12: **end for**
13: **end for**
//Phase 2: Payment Determination
14: **foreach** $u_j \in U$ **do**
15: $p_j \leftarrow \infty$;
16: **end for**
17: **foreach** $u_j \in S$ **do**
18: $U' \leftarrow U \backslash \{u_j\}$; $S' \leftarrow \emptyset$; $\mathbf{D}'' \leftarrow \mathbf{D}$;
19: sort all users in based on $\frac{b_j}{\sum_{i\in R} a_j^i}$ for $\forall u_j \in U'$ in nonincreasing order and the sequence is denoted by $Q_{U'}$;
20: **foreach** $j_h \in Q_{U'}$ in order **do**
21: **foreach** $e_k \in Q_E$ in order **do**
22: **if** $d''^i_k \geq a^i_{jh}, \forall i \in R$ **then**
23: $S' \leftarrow S' \cup \{j_h\}$;
24: $p_j \leftarrow \min\{p_j, \frac{\sum_{i\in R} a_j^i}{\sum_{i\in R} a^i_{jh}} b_{jh}\}$;
25: **end if**
26: **foreach** $i \in R$ **do**
27: $d''^i_k \leftarrow d''^i_k - a^i_{jh}$;
28: **end for**
29: **end for**
30: **end for**
31: **end for**

In payment determination phase, for each winner $u_j \in S$, we execute the winner selection phase over $U \backslash \{u_j\}$, and the winner set is denoted by S'. We compute the minimum price that user u_j can be selected instead of each user in S'(Line 24). We will prove that this price is a critical payment for user u_j later.

4.2. Auction analysis

In this subsection, we present the theoretical analysis, demonstrating that the global auction can achieve the desired properties.

Lemma 1. *The global auction is computationally efficient.*

Proof. Sorting the user set (Line 2) takes $O(nrlogn)$. Sorting the MEC set (Line 3) takes $O(mrlogm)$. The task allocation (Lines 4–13) takes $O(nmr)$. Since $n > m$, the time complexity of winner selection is $\max\{nrlogn, nmr\}$. In each iteration of the for-loop (Lines 17–31), a process similar to Lines 4–13 is executed. Hence the time complexity of the whole auction is dominated by this for-loop, which is bounded by $O(n^2mr)$. ■

Lemma 2. *The global auction is individually rational.*

Proof. Let j_h be user u_j's replacement which appears in the jth place in the sorting over $U \backslash \{u_j\}$. Since user j_h would not be at jth place if j is considered, we have $\frac{b_j}{\sum_{i\in R} a_j^i} \geq \frac{b_{j_h}}{\sum_{i\in R} a_{j_h}^i}$. Hence $b_j \geq \frac{\sum_{i\in R} a_j^i}{\sum_{i\in R} a_{j_h}^i} b_{i_h}$. This is sufficient to guarantee $b_j \geq min_{j_h \in S'} \frac{\sum_{i\in R} a_j^i}{\sum_{i\in R} a_{j_h}^i} b_{j_h} = p_j$. ■

Before analyzing the truthfulness of global auction, we firstly introduce the Myerson's Theorem [30].

Theorem 2. *An auction is truthful if and only if:*

- *The selection rule is monotone: If user u_j wins the auction by bidding b_j, it also wins by bidding $b_j' \geq b_j$;*
- *Each winner needs to pay the critical value: User u_j would not win the auction if it bids lower than this value.*

Lemma 3. *The global auction is truthful.*

Proof. Based on Theorem 2, it suffices to prove that the selection rule of global auction is monotone and the payment p_j for each user u_j is the critical value. The monotonicity of the selection rule is obvious as bidding a larger value cannot push user u_j backwards in the sorting. We next show that p_j is the critical value for u_j in the sense that bidding smaller p_j could prevent u_j from winning the auction. Note that $p_j = min_{h\in\{1,\ldots,L\}} \frac{\sum_{i\in R} a_j^i}{\sum a_{j_h}^i} b_{j_h}$, where L is the position of last winner in payment determination phase. If user j bid $b_j \leq p_j$, it will be placed after L since $b_j \leq \frac{\sum_{i\in R} a_j^i}{\sum_{i\in R} a_{j_L}^i} b_{j_L}$ implies $\frac{b_j}{\sum_{i\in R} a_j^i} \leq \frac{b_{j_L}}{\sum_{i\in R} a_{j_L}^i}$. Hence, user u_j would not win the auction because the first L users have offloaded the tasks to MECs, and the remaining resources cannot satisfy any task after position L. ■

According to Theorem 4.3 in [28], the simple approach that greedily select a single multi-dimensional knapsack to pack until all of the knapsacks are filled yields a guaranteed approximation. Based on Theorem 1, the *GSOTO* problem given in formula (4) is an instance of *MMDK* problem, and our Global Auction (Algorithm 1) follows this greedy algorithm. Specifically, we sort all users based on the unit resource value $\frac{b_j}{\sum_{i\in R} a_j^i}$ for all $u_j \in U$ in nonincreasing order and offload the current highest unit resource value items, one by one, into the MEC with minimum resource capacity under the resource capacity constraints. Thus, we have

Lemma 4. *The global auction can approximate the optimal solution within a factor of $\frac{1}{O(2H_K)}$, where $K = max_{u_j\in U} \sum_{i\in R} a_j^i$, H_K is the first K terms of the harmonic series.*

The above four lemmas together prove the following theorem.

Theorem 3. *The global auction is computationally efficient, individually rational, truthful, and $\frac{1}{O(2H_K)}$ approximate, where $K = max_{u_j\in U} \sum_{i\in R} a_j^i$.*

5. Collaborative task offloading in secondary auction-based heterogeneous MEC model

In this section, we present the local auction and secondary auction in secondary auction-based heterogeneous MEC model.

5.1. Auction design

We also attempt to find efficient algorithm for the *LSOTO* problem presented in (7). Unfortunately, as shown in the following theorem, it is NP-hard to find the optimal solution.

Theorem 4. *The LSOTO problem is NP-hard.*

Proof. We consider a corresponding instance of *LSOTO*: Let $T^k = \{t_1, t_2, \dots, t_{|T^k|}\}$ denote the set of tasks of any MEC e_k. Each task $t_j \in T^k$ has a resource requirement $\boldsymbol{a_j} = (a_j^1, a_j^2, \dots, a_j^r)$ and a value v_j. The MEC e_k is with resource capacity $\boldsymbol{d_k} = (d_k^1, d_k^2, \dots, d_k^r)$. The target is to maximize the sum of the values of the tasks assigned to e_k so that the sum of resource requirement of tasks in e_k for every resource $i \in \{1, 2, \dots, r\}$ does not exceed the resource capacity d_k^i of MEC e_k. Then we can see that this problem is the *Multi-Dimension Knapsacks (MDK)* problem. Since the *MDK* problem is a well-known NP-hard problem, the *LSOTO* problem is NP-hard. ■

As illustrated in Algorithm 2, the local auction in each MEC e_k follows a greedy approach consisting of winner selection phase and payment determination phase.

In the winner selection phase, the users are sorted according to the unit resource value, which is defined as $\frac{b_j}{\sum_{i\in R} a_j^i}$ for any user $u_j \in U^k$, in nonincreasing order (Line 2). We iterate all users in order, and try to allocate the user to the MEC in each iteration. If the resource capacity of the MEC can satisfy the resource requirement of the user, the user is selected as a winner (Lines 4–6).

In payment determination phase, for each winner $u_j \in S^k$, we execute the winner selection phase over $U^k \backslash \{u_j\}$, and the winner set is denoted by S'^k. We compute the minimum price that user u_j can be selected instead of each user in S'^k (Line 20). We will prove that this price is a critical payment for user u_j later.

Algorithm 2: Local Auction

Input: user set U^k, bid profile $\mathbf{B^k}$ and resource capacity $\boldsymbol{d_k}$
Output: winner set S^k, payment profile $\mathbf{p^k}$
//Phase 1: Winner Selection
1: $S^k \leftarrow \emptyset$; $\boldsymbol{d'_k} \leftarrow \boldsymbol{d_k}$;
2: sort all users based on $\frac{b_j}{\sum_{i\in R} a_j^i}$ for $\forall u_j \in U^k$ in nonincreasing order and the sequence is denoted by Q_{U^k};
3: **foreach** $u_j \in Q_U^k$ in order **do**
4: **if** $d'^i_k \geq a_j^i, \forall i \in R$ **then**
5: $S^k \leftarrow S^k \cup \{u_j\}$;
6: **foreach** $i \in R$ **do**
7: $d'^i_k \leftarrow d'^i_k - a_j^i$;
8: **end for**
9: **end if**
10: **end for**
//Phase 2: Payment Determination
11: **foreach** $u_j \in U^k$ **do**
12: $p_j \leftarrow \infty$;
13: **end for**
14: **foreach** $u_j \in S^k$ **do**
15: $U'^k \leftarrow U^k \backslash \{u_j\}$; $S'^k \leftarrow \emptyset$; $\boldsymbol{d''_k} \leftarrow \boldsymbol{d_k}$;
16: sort all users in based on $\frac{b_j}{\sum_{i\in R} a_j^i}$ for $\forall u_j \in U'^k$ in nonincreasing order and the sequence is denoted by $Q_{U'^k}$;
17: **foreach** $j_h \in Q_{U'^k}$ in order **do**
18: **if** $d''^i_k \geq a_{jh}^i, \forall i \in R$ **then**
19: $S'^k \leftarrow S'^k \cup \{j_h\}$;
20: $p_j \leftarrow \min\{p_j, \frac{\sum_{i\in R} a_j^i}{\sum_{i\in R} a_{jh}^i} b_{jh}\}$;
21: **end if**
22: **foreach** $i \in R$ **do**
23: $d''^i_k \leftarrow d''^i_k - a_{jh}^i$;
24: **end for**
25: **end for**
26: **end for**

The algorithm of secondary auction is very similar to the local auction. The difference is that the busy MEC will deduct the agency fee after it receives the payment from users.

5.2. Auction analysis

In this subsection, we present the theoretical analysis, demonstrating that the both local auction and secondary auction can achieve the desired properties.

Theorem 5. *The local auction is computationally efficient, individually rational, truthful, and* $\frac{1}{O(H_K)}$ *approximate, where* $K = max_{u_j\in U^k} \sum_{i\in R} a_j^i$.

Proof. For the time complexity, sorting the user set (Line 2) takes *O(nrlogn)*. The task allocation (Lines 3–10) takes *O(nr)*. In each iteration of the for-loop (Lines 14–26), a process similar to Lines 3–10 is executed. Hence the time complexity of the whole auction is dominated by this for-loop, which is bounded by $O(n^2 r)$.

The proofs for individual rationality and truthfulness of users are similar to those of global auction.

For *MDK* problem, the best-known approximation algorithm is to greedily add items with the best bang-for-bucks without exceeding the capacity of knapsack. The bang-for-buck of item is defined as the ratio of value to the total weights of all dimensions. According to Theorem 4.1 in [28], the greedy algorithm that sorts items according to the bang-for-buck and packs the current highest bang-for-buck items, one by one, into the knapsack without violating the capacity constraints yields a guaranteed approximation. Based on Theorem 4, the *LSOTO* problem given in formula (7) is an instance of *MDK* problem, and our Local Auction (Algorithm 2) follows this greedy algorithm. Specifically, we sort all users based on the unit resource value $\frac{b_j}{\sum_{i\in R} a_j^i}$ for any user $u_j \in U^k$ in nonincreasing order and offload the current highest unit resource value items, one by one, into the MEC without violating the resource capacity constraints. Thus, the approximation can be obtained from Theorem 4.1 in [28]. ■

Given the bidder set $U \backslash \bigcup_{e_k\in E} S^k$ of secondary auction, i.e., the loser set of local auctions, we can also have the following theorem.

Theorem 6. *The secondary auction is computationally efficient, individually rational, truthful, and* $\frac{1}{O(H_K)}$ *approximate, where* $K = max_{u_j\in U\backslash \bigcup_{e_k\in E} S^k} \sum_{i\in R} a_j^i$.

Remark. Note that the running time of local auction and secondary auction is very conservative since the number of both users and winners is much less than n in practice. Specifically, the user set of local auction and secondary auction are U^k and $U \backslash \bigcup_{e_k\in E} S^k$, respectively. The size of both of them is much smaller than n.

6. Collaborative task offloading in double auction-based heterogeneous MEC model

Since the local auction has been presented in Section 5, we only present the double auction in this section.

6.1. Auction design

Actually, the double auction is a combinatorial double auction since each user requires an indivisible bundle of r resources. As illustrated in Algorithm 3, we design the double auction through the combination of well-known McAfee double auction [29] and the global auction presented in Algorithm 1.

First, we calculate the unit resource bid price f_j of each user. Then we sort users based on the unit resource bid price in nonincreasing order (Line 5) and sort idle MECs based on unit resource ask price (Line 6). We find the last position l such that $f_l \geq g_l$ (Line 7). We consider the following two cases:

Case 1: If $\frac{f_{l+1} + g_{l+1}}{2} \in [g_l, f_l]$, the first l users and idle MECs are selected as candidates, and we put them into candidate winning user set S_c^U (Line 9) and candidate winning idle MEC set S_c^E (Line 10), respectively. We denote $\boldsymbol{B}_c$ and $\boldsymbol{D}_c$ as the bid profile of candidate winning users and resource capacity matrix, respectively. Then we calculate the winning user set S^U, the payment profile of users $\boldsymbol{p}^G$ and resource allocation profile $\mathbf{q}$ by calling global auction (Line 11), which has been given in Algorithm 1. For each user $u_j \in S^U$, we compute the maximal price of p_j^G and $\frac{f_{l+1}+g_{l+1}}{2}\sum_{i\in R} a_j^i$ (Line 13). We will prove that this price is a critical payment for user j later. Meanwhile, we can determine the wining idle MEC set based on resource allocation profile $\mathbf{q}$, and the payment for each $e_k \in S^E$ is $\frac{f_{l+1}+g_{l+1}}{2}\sum_{u_j\in S^U,\ q_j=e_k}\sum_{i\in R} a_j^i$ (Line 19).

Case 2: If $\frac{f_{l+1}+g_{l+1}}{2} \in [g_l, f_l]$, the first $l-1$ users and idle MECs are selected as candidates. Different from Case 1, the payment for any winning user $u_j \in S^U$ is the maximal price of p_j^G and $f_l \sum_{i\in R} a_j^i$ (Line 26). The payment for each wining idle MEC $e_k \in S^E$ is $g_l \sum_{u_j\in S^U,\ q_j=e_k}\sum_{i\in R} a_j^i$ (Line 32).

Algorithm 3: Double Auction

Input: user set U^L, idle MEC set E^{idle}, bid profile $\mathbf{B^L}$ and resource capacity $\mathbf{A^{idle}}$
Output: winner set S^U, winning idle MEC set S^E, resource allocation profile $\mathbf{q}$, payment profile $\mathbf{p}$

1: $S^U \leftarrow \emptyset$; $S^E \leftarrow \emptyset$; $\mathbf{q} \leftarrow 0$; $\mathbf{p} \leftarrow 0$;
2: **foreach** $u_j \in U^L$ **do**
3: $\quad f_j \leftarrow \frac{b_j}{\sum_{i\in R} a_j^i}$;
4: **end for**
5: sort all users based on f_j for $\forall u_j \in U^L$ in nonincreasing order and the sequence is denoted by Q_{U^L};
6: sort all idle MECs based on g_k for $\forall e_k \in E^{idle}$ in nondecreasing order and the sequence is denoted by $Q_{E^{idle}}$;
7: find the last position l such that $f_l \ge g_l$
8: **if** $\frac{f_{l+1}+g_{l+1}}{2} \in [g_l, f_l]$ **then**
9: $\quad$ let S_c^U be the set of first l users in Q_{U^L};
10: $\quad$ let S_c^E be the set of first l idle MECs in $Q_{E^{idle}}$;
11: $\quad$ $(S^U, \boldsymbol{p}^G, \mathbf{q}) \leftarrow$ **Global Auction**$(S_c^U, S_c^E, \boldsymbol{B}_c, \boldsymbol{D}_c)$;
12: $\quad$ **foreach** $u_j \in S^U$ **do**
13: $\quad\quad p_j^U \leftarrow \max\{p_j^G, \frac{f_{l+1}+g_{l+1}}{2}\sum_{i\in R} a_j^i\}$;
14: $\quad$ **end for**
15: $\quad$ **foreach** $u_j \in S^U$ **do**
16: $\quad\quad S^E \leftarrow S^E \cup \{q_j\}$;
17: $\quad$ **end for**
18: $\quad$ **foreach** $e_k \in S^E$ **do**
19: $\quad\quad p_k^E \leftarrow \frac{f_{l+1}+g_{l+1}}{2}\sum_{u_j\in S^U,\ q_j=e_k}\sum_{i\in R} a_j^i$;
20: $\quad$ **end for**
21: **else if** $\frac{f_{l+1}+g_{l+1}}{2} \notin [g_l, f_l]$ **then**
22: $\quad$ let S_c^U be the set of first $l-1$ users in Q_{U^L};
23: $\quad$ let S_c^E be the set of first $l-1$ idle MECs in $Q_{E^{idle}}$;
24: $\quad$ $(S^U, \boldsymbol{p}^G, \mathbf{q}) \leftarrow$ **Global Auction**$(S_c^U, S_c^E, \boldsymbol{B}_c, \boldsymbol{D}_c)$;
25: $\quad$ **foreach** $u_j \in S^U$ **do**
26: $\quad\quad p_j^U \leftarrow \max\{p_j^G, f_l \sum_{i\in R} a_j^i\}$;
27: $\quad$ **end for**
28: $\quad$ **foreach** $u_j \in S^U$ **do**
29: $\quad\quad S^E \leftarrow S^E \cup \{q_j\}$;
30: $\quad$ **end for**
31: $\quad$ **foreach** $e_k \in S^E$ **do**
32: $\quad\quad p_k^E \leftarrow g_l \sum_{u_j\in S^U,\ q_j=e_k}\sum_{i\in R} a_j^i$;
33: $\quad$ **end for**
34: **end if**

6.2. Auction analysis

In this subsection, we present the theoretical analysis, demonstrating that the double auction can achieve the desired properties.

Lemma 5. *The double auction is computationally efficient.*

Proof. The running time of double auction is dominated by the global auction, which is $O(n^2mr)$ given in Lemma 1.

Remark. Note that the running time of double auction is very conservative since the number of both users and winners is much smaller than n, and the number of idle MECs is less than m in practice. ■

Lemma 6. *The double auction is individually rational.*

Proof. We first analyze the utility of users.

For Case 1: the payment for user u_j is $p_j^U = \max\{p_j^G, \frac{f_{l+1}+g_{l+1}}{2}\sum_{i\in R} a_j^i\}$ based on Line 13. Note that p_j^G is the critical value of payment in global auction. Thus we have $p_j^G \ge b_j$ based on Lemma 2. Since $\frac{f_{l+1}+g_{l+1}}{2} \in [g_l, f_l]$ in this case, we have $\frac{f_{l+1}+g_{l+1}}{2} \le f_l \le f_j$, where the last inequation relies on $j \le l$. Then, we have $\frac{f_{l+1}+g_{l+1}}{2}\sum_{i\in R} a_j^i \le f_j \sum_{i\in R} a_j^i = b_j$. Thus, we have $p_j^U \le b_j$.

For Case 2: $p_j^U = \max\{p_j^G, f_l \sum_{i\in R} a_j^i\}$ based on Line 26. Obviously, $p_j^G \le b_j$ based on Lemma 2. Since $j \le l$ in this case, we have $f_l \le f_j$. Then, we have $f_l \sum_{i\in R} a_j^i \le f_j \sum_{i\in R} a_j^i = b_j$. Thus, we have $p_j^U \le b_j$.

Next, we analyze the utility of idle MECs.

For Case 1: the payment for idle MEC e_k is $p_k^E = \frac{f_{l+1}+g_{l+1}}{2}\sum_{u_j\in S^U,\ q_j=e_k}\sum_{i\in R} a_j^i$ based on Line 19. Since $\frac{f_{l+1}+g_{l+1}}{2} \in [g_l, f_l]$ in this case, we have $\frac{f_{l+1}+g_{l+1}}{2} \ge g_l \ge g_k$, where the last inequation relies on $k \le l$. Thus, we have $p_k^E \ge g_k \sum_{u_j\in S^U,\ q_j=e_k}\sum_{i\in R} a_j^i$.

For Case 2: the payment for idle MEC e_k is $p_k^E = g_l \sum_{u_j\in S^U,\ q_j=e_k}\sum_{i\in R} a_j^i$ based on Line 32. Since $k < l$ in this case, we have $g_k \le g_l$. Thus, we have $p_k^E \ge g_k \sum_{u_j\in S^U,\ q_j=e_k}\sum_{i\in R} a_j^i$. ■

Lemma 7. *The double auction satisfies budget balance.*

Proof. For Case 1: the payment for each user $u_j \in S^U$ is $p_j^U = \max\{p_j^G, \frac{f_{l+1}+g_{l+1}}{2}\sum_{i\in R} a_j^i\} \ge \frac{f_{l+1}+g_{l+1}}{2}\sum_{i\in R} a_j^i$. Thus the unit resource payment is no less than $\frac{f_{l+1}+g_{l+1}}{2}$. On the other hand, the payment for each idle MEC $\forall e_k \in S^E$ is $p_k^E = \frac{f_{l+1}+g_{l+1}}{2}\sum_{u_j\in S^U,\ q_j=e_k}\sum_{i\in R} a_j^i$ with unit resource payment $\frac{f_{l+1}+g_{l+1}}{2}$. Thus, we have $\sum_{u_j\in S^U} p_j^U - \sum_{e_k\in S^E} p_k^E$.

For Case 2: the payment for each user $u_j \in S^U$ is $p_j^U = \max\{p_j^G, f_l \sum_{i\in R} a_j^i\} \ge f_l \sum_{i\in R} a_j^i$. Thus the unit resource payment is no less than f_l. On the other hand, the payment for each idle MEC $\forall e_k \in S^E$ is $p_k^E = g_l \sum_{u_j\in S^U,\ q_j=e_k}\sum_{i\in R} a_j^i$ with unit resource payment g_l. Since $g_l \le f_l$, we have $\sum_{u_j\in S^U} p_j^U - \sum_{e_k\in S^E} p_k^E$.

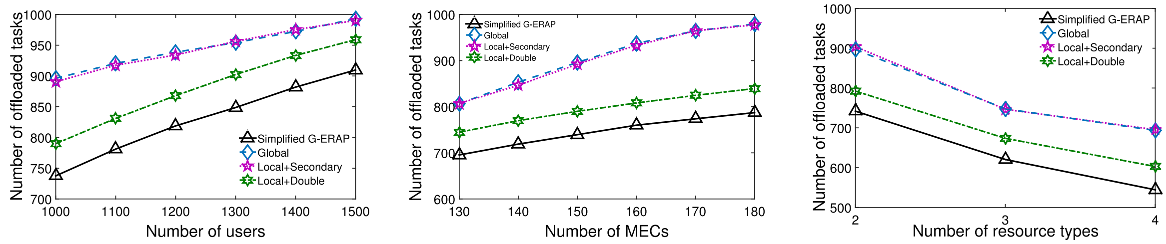
Lemma 8. *The double auction is truthful.*

Proof. We prove the truthfulness based on Theorem 2. We first show that the users are truthful.

Case 1: $\frac{f_{l+1}+g_{l+1}}{2} \in [g_l, f_l]$. In this case, $S_c^U = \{1, 2, \dots, l\}$. For any $S_c^U \in S^U$, the monotonicity of the candidate selection rule is obvious as bidding a larger value cannot push user u_j backwards in the sorting of Line 5 of Algorithm 3. In the global auction, the monotonicity of the winner selection rule is also obvious since bidding a larger value cannot push user u_j backwards in the sorting of Line 2 of Algorithm 1. We next show that p_j^U is the critical value for u_j in the sense that bidding smaller p_j^U could prevent u_j from winning the double auction.

To ensure that j is a candidate, the following inequations should be held:

$$f_j \ge f_{l+1} \tag{14}$$



(a) (b) (c)

Fig. 5. Number of offloaded tasks. (a) Impact of number of users. (b) Impact of number of MECs. (c) Impact of number of resource types.

$$f_j \geq g_l \tag{15}$$

$$f_j \geq \frac{f_{l+1}+g_{l+1}}{2} \tag{16}$$

Since $g_{l+1} > f_{l+1}$, we have $\frac{f_{l+1}+g_{g+1}}{2} > f_{l+1}$. In addition, $\frac{f_{l+1}+g_{l+1}}{2} \in [g_l, f_l]$ in this case, we have $\frac{f_{l+1}+g_{l+1}}{2} \geq g_l$. Thus (14), (15) and (16) can be simplified as: $f_j \geq \frac{f_{l+1}+g_{l+1}}{2}$, i.e., $\frac{f_{l+1}+g_{l+1}}{2} \sum_{i\in R} a_j^i$ is the critical value to ensure that u_j is a candidate. Note that p_j^G is the critical value of being the winner in the global auction based on Lemma 3. So, p_j^U, which is the maximum of two critical values in candidate section and global auction, is the critical value of the whole double auction.

Case 2: $\frac{f_{l+1}+g_{l+1}}{2} \notin [g_l, f_l]$. In this case, $S_c^U = \{1, 2, \ldots, l-1\}$. The proof of monotonicity is same with Case 1. To ensure that j is a candidate, the following inequations should be held:

$$f_j \geq f_l \tag{17}$$

$$f_j \geq g_{l-1} \tag{18}$$

Since $f_l \geq g_l \geq g_{l-1}$, $f_l \sum_{i\in R} a_j^i$ is the critical value to assure j of being a candidate. The rest of the proof is same with Case 1.

Next, we show that the idle MECs are truthful.

Case 1: $\frac{f_{l+1}+g_{l+1}}{2} \in [g_l, f_l]$. In this case, $S_c^E = \{e_1, e_2, \ldots, e_l\}$. For any $e_k \in S^E$, the monotonicity of the candidate selection rule is obvious as asking a small cost cannot push user u_j backwards in the sorting of Line 6 of Algorithm 3. The global auction is not related to the ask price. Thus, both candidate selection rule and global auction are monotone. We next show that p_k^E is the critical value for e_k in the sense that asking higher p_k^E could prevent e_k from winning the double auction.

To ensure that e_k is a candidate, the following in equations should be held:

$$g_k \leq g_{l+1} \tag{19}$$

$$g_k \leq f_l \tag{20}$$

$$g_k \leq \frac{f_{l+1}+g_{l+1}}{2} \tag{21}$$

Since $f_{l+1} < g_{l+1}$, we have $\frac{f_{l+1}+g_{l+1}}{2} < g_{l+1}$. In addition, $\frac{f_{l+1}+g_{l+1}}{2} \in [g_l, f_l]$ in this case, we have $\frac{f_{l+1}+g_{l+1}}{2} \leq f_l$. Thus (19), (20) and (21) can be simplified as: $g_k \leq \frac{f_{l+1}+g_{l+1}}{2}$, i.e., $p_k^E = \frac{f_{l+1}+g_{l+1}}{2} \sum_{u_j\in S^U,\ q_j=e_k} \sum_{i\in R} a_j^i$ is the critical value of the whole double auction.

Case 2: $\frac{f_{l+1}+g_{l+1}}{2} \notin [g_l, f_l]$. In this case, $S_c^E = \{e_1, e_2, \ldots, e_{l-1}\}$. The proof of monotonicity is same with Case 1. To ensure that e_k is a candidate, the following inequations should be held:

$$g_k \leq g_l \tag{22}$$

$$g_k \leq f_{l-1} \tag{23}$$

Since $f_{l-1} \geq f_l \geq g_l$, $p_k^E = g_l \sum_{u_j\in S^U,\ q_j=e_k} \sum_{i\in R} a_j^i$ is the critical value of the whole double auction. ■

The above four lemmas together prove the following theorem.

Theorem 7. *The double auction is computationally efficient, individually rational, budget-balanced and truthful*

7. Performance evaluation

We have conducted simulations to investigate the performance of global auction, secondary auction, double auction and Simplified G-ERAP algorithm [31] based on the real experience dataset.

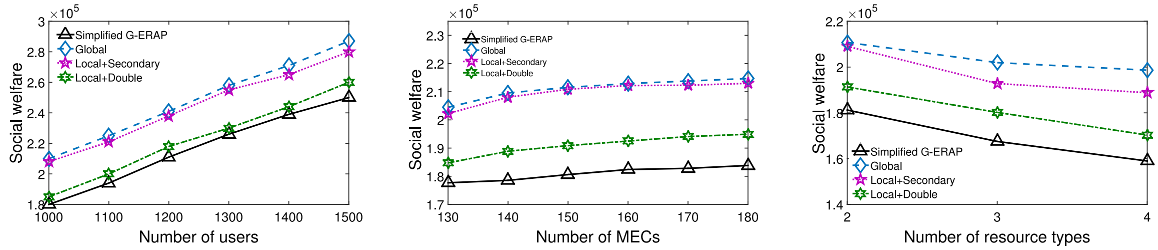
7.1. Simulation setup

We extract resource requirement of tasks and resource capacity of MECs from the Alibaba Open Cluster Trace Program [32], which contains over a million of tasks and 4034 machines in a period of 8 days. We choose a subset of this dataset for our simulations. The default values of the parameters are as follows: $m = 150, n = 1000$ and $r = 2$. We will vary the value of the key parameters to explore the impacts on designed auctions. The tasks are randomly distributed to local MECs. The value of each task is selected randomly form the auction dataset, which contains 5017 bid prices for Palm Pilot M515 PPD from eBay [33].

In our simulations, we measure the number of offloaded tasks, social welfare, and running time with different number of users (n), number of MECs (m), and number of resources (r) of global auction for homogeneous MECs and dual auction (local auction + secondary auction, and local auction + double auction) for cooperative task offloading in heterogeneous MECs. We use Simplified G-ERAP algorithm [31], which greedy allocates tasks based on the average bid per unit of resource, as the comparison algorithm. Note that the global auction actually provides the performance upper bound for heterogeneous MEC task offloading. Then we verify the truthfulness of proposed auctions. All the simulations were run on a Windows 7 machine with Intel(R) Xeon(R) CPU I5-3230M 2.6 GHz and 8 GB memory. Each measurement is averaged over 100 instances.

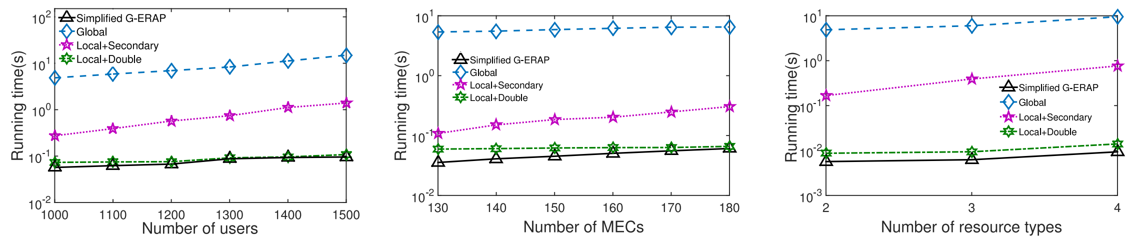
7.2. Number of offloaded tasks

We first investigate the number of offloaded tasks of all auctions. We can see from Fig. 5 that the number of offloaded tasks of all auctions increases with the increasing number of total users. The secondary auction offloads 82.4% of all tasks on average, and obtains most offloaded tasks of the auctions for heterogeneous MECs. The Simplified G-ERAP algorithm offloads only 69% of all tasks since it cannot offload the tasks to remote idle MECs. Compared with Simplified G-ERAP algorithm, the double auction offloads 6.4% more tasks averagely. Although the



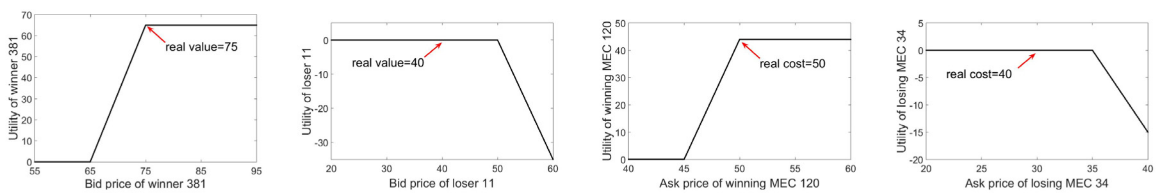
(a) (b) (c)

Fig. 6. Social welfare. (a) Impact of number of users. (b) Impact of number of MECs. (c) Impact of number of resource types.



(a) (b) (c)

Fig. 7. Running time. (a) Impact of number of users. (b) Impact of number of MECs. (c) Impact of number of resource types.



(a) (b) (c) (d)

Fig. 8. Truthfulness of double auction. (a) Utility of winner 381. (b) Utility of loser 11. (c) Utility of winning MEC 120. (d) Utility of losing MEC 34.

double auction offloads fewer tasks than secondary auction, it guarantees the truthfulness for both users and idle MECs (will be verified in Section 7.5) and is very computationally efficient (will be verified in Section 7.4). All auctions offload more tasks when there are more MECs since there are more available resources. With more types of resources, it is harder to meet the requirements of tasks, thus the number of offloaded tasks of all auctions decreases.

7.3. Social welfare

Then, we investigate the social welfare of all auctions. Since all proposed auctions are truthful, maximizing the social welfare is equivalent to maximizing the total values of offloaded tasks. In addition, the proposed auctions except global auction are greedy algorithms. Thus, the social welfare largely depends on the number of offloaded tasks. As shown in Fig. 6, the social welfare of all auctions increases when the number of users or the number of MECs increases. This is because that more tasks will be offloaded in these two cases. The social welfare of all auctions decreases when there are more types of resources. Overall, secondary auction can obtain 98.5% social welfare of global auction averagely. Moreover, the secondary auction and double auction can obtain 14.5% and 4.2% more social welfare than comparison algorithm on average, respectively.

7.4. Running time

Further, we measure the running time of all auctions. As shown in Fig. 7, the running time of all auctions increases when the number of users, number of MECs, or number of resource types increases. The global auction is the slowest of all auctions since it has the most bidders. The secondary auction and double auction are more efficient than global auction since the number of bidders is much smaller than that of global auction (see our analysis in Section 5.2 and 6.2, respectively). For the setting $n = 1500, m = 150$, our secondary auction and double auction only take 6.5% and 0.85% of running time compared with global auction, respectively. We can see that the double auction shows great advantage in terms of computation efficiency since it only happens in single busy MEC.

7.5. Truthfulness

To avoid redundancy, we only verify the truthfulness of double auction by randomly picking two users and two MECs and allowing them to bid/ask prices that are different from their true values/costs. We change the bid/ask prices of picked users or MECs while others' bid/ask prices are not changed, and observe the utilities with different bid/ask prices. We illustrate the results in Fig. 8. We can see that the winner 381 always obtain its maximum utility of 65 if bidding its real value 75. Accordingly, the loser 11 always obtains nonnegative utility if he/she bids truthfully. We also see that the winning MEC 120 always obtain its maximum utility of 44 if bidding its real cost 50. Accordingly, the losing MEC 34 always obtains nonnegative utility if he/she bids truthfully.

8. Conclusion

In this paper, we have designed truthful auctions for the heavily loaded task offloading system in both homogeneous MECs and heterogeneous MECs. We have designed the system models and formulated the social optimal task offloading problem for these two scenarios. For the homogeneous MECs, we have presented a global auction executed in the central cloud as the benchmark of the auctions proposed in heterogeneous MEC situation. For the heterogeneous MECs, we have designed two dual auction models: secondary auction-based model and double auction-based model. We have designed the auctions for these two dual auction models. We have demonstrated that the proposed auctions achieve desirable properties of computational efficiency, individual rationality, budget balance, truthfulness, and guaranteed approximation. The simulation results have shown that the designed secondary auction can obtain 98.5% social welfare of global auction averagely, and the proposed double auction has great advantage in terms of computation efficiency.

CRediT authorship contribution statement

Weifeng Lu: Conceptualization, Methodology. **Weiduo Wu:** Data curation, Software, Writing – original draft. **Jia Xu:** Methodology, Supervision. **Pengcheng Zhao:** Visualization, Investigation. **Dejun Yang:** Supervision. **Lijie Xu:** Software, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Mumtaz, A. Bo, A. Al-Dulaimi, K. Tsang, Guest editorial 5G and beyond mobile technologies and applications for industrial IoT (IIoT), IEEE Trans. Ind. Inf. 14 (61) (2018) 2588–2591.

[2] J. Wu, R. Tan, M. Wang, Streaming high-definition real-time video to mobile devices with partially reliable transfer, IEEE Trans. Mob. Comput. 18 (2) (2019) 458–472.

[3] C. Tiffany Yu-Han, Ravindranathm L., S. Deng, P. Bahl, H. Balakrishnan, Glimpse: Continuous, real-time object recognition on mobile devices, in: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, 2015, pp. 155–168.

[4] T. Tran, A. Hajisami, P. Pandey, D. Pompili, Collaborative mobile edge computing in 5g networks: new paradigms, scenarios, and challenges, IEEE Commun. Mag. 55 (4) (2017) 54–61.

[5] J. Xue, Q. Hu, Y. An, L. Wang, Joint task offloading and resource allocation in vehicle-assisted multi-access edge computing, Comput. Commun. 177 (2021) 77–85.

[6] C.F. Liu, M. Bennis, M. Debbah, H.V. Poor, Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing, IEEE Trans. Commun. 67 (6) (2019) 4132–4150.

[7] X. Chen, L. Jiao, W. Li, X. Fu, Efficient multi-user computation offloading for mobile-edge cloud computing, IEEE/ACM Trans. Netw. 24 (5) (2016) 2795–2808.

[8] J. Xu, Z. Ren, L. Xu, D. Yang, T. Li, Incentive mechanism for multiple cooperative tasks with compatible users in mobile crowd sensing via online communities, IEEE Trans. Mob. Comput. 19 (7) (2020) 1618–1633.

[9] L. Jiang, X. Niu, J. Xu, D. Yang, L. Xu, Incentive mechanism design for truth discovery in crowdsourcing with copiers, IEEE Trans. Serv. Comput. (2021) http://dx.doi.org/10.1109/TSC.2021.3075741.

[10] J. Xu, H. Wu, W. Lu, L. Xu, Incentive mechanism for rational miners in Bitcoin mining pool, Inf. Syst. Front. 23 (2) (2021) 317–327.

[11] J. Xu, Y. Zhou, Y. Ding, D. Yang, L. Xu, Bi-objective robust incentive mechanism design for mobile crowdsensing, IEEE Internet Things J. (2021) http://dx.doi.org/10.1109/JIOT.2021.3072953.

[12] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, H. Zhang, Incentive mechanism for computation offloading using edge computing: A stackelberg game approach, Comput. Netw. 129 (2017) 399–409.

[13] G. Li, J. Cai, An online incentive mechanism for collaborative task offloading in mobile edge computing, IEEE Trans. Wireless Commun. 19 (1) (2020) 624–636.

[14] D. Zhang, L. Tan, J. Ren, M. Khattar Awad, S. Zhang, Y. Zhang, P. Wan, Near-optimal and truthful online auction for computation offloading in green edge-computing systems, IEEE Trans. Mob. Comput. 19 (4) (2020) 880–893.

[15] S. Josilo, G. Dan, A game theoretic analysis of selfish mobile computation offloading, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, 2017.

[16] Chen. M, Y. Hao, Task offloading for mobile edge computing in software defined ultra-dense network, IEEE J. Sel. Areas Commun. 36 (3) (2018) 587–597.

[17] Y. Zeng, Y. Huang, Z. Liu, Y. Yang, Joint online edge caching and load balancing for mobile data offloading in 5G networks, in: 2019 IEEE 39th International Conference on Distributed Computing Systems, ICDCS, 2019.

[18] G. Li, Y. Yao, J. Wu, X. Liu, Q. Lin, A new load balancing strategy by task allocation in edge computing based on intermediary nodes, EURASIP J. Wireless Commun. Networking 1 (3) (2020) 1–10.

[19] A. Rahman, J. Jin, A. Rahman, A. Cricenti, M. Afrin, Energy-efficient optimal task offloading in cloud networked multi-robot systems, Comput. Netw. 160 (4) (2019) 11–32.

[20] S. Guo, J. Liu, Y. Yang, B. Xiao, Z. Li, Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing, IEEE Trans. Mob. Comput. 18 (2) (2019) 319–333.

[21] J. Luis, J. Laredo, F. Guinand, D. Olivier, P. Bouvry, Load balancing at the edge of chaos: how self-organized criticality can lead to energy-efficient computing, IEEE Trans. Parallel Distrib. Syst. 28 (2) (2016) 517–529.

[22] J. Xu, J. Xiang, D. Yang, Incentive mechanisms for time window dependent tasks in mobile crowdsensing, IEEE Trans. Wireless Commun. 14 (11) (2015) 6353–6364.

[23] J. Xu, J. Xiang, Y. Li, Incentivize maximum continuous time interval coverage under budget constraint in mobile crowd sensing, Wirel. Netw. 36 (5) (2017) 1549–1562.

[24] X. Tan, A. Leon-Garcia, Y. Wu, D.H.K. Tsang, Online combinatorial auctions for resource allocation with supply costs and capacity limits, IEEE J. Sel. Areas Commun. 38 (4) (2020) 655–668.

[25] P. Samimi, Y. Teimouri, M. Mukhtar, A combinatorial double auction resource allocation model in cloud computing, Inform. Sci. 357 (2016) 201–216.

[26] U. Habiba, S. Maghsudi, E. Hossain, Reverse auction model for efficient resource allocation in mobile edge computation offloading, in: 2019 IEEE Global Communications Conference, GLOBECOM, 2019.

[27] R. Zhang, W. Shi, J. Zhang, W. Liu, An auction scheme for computing resource allocation in D2D-assisted mobile edge computing, in: IEEE Global Communications Conference, GLOBECOM, 2019.

[28] H. Chan, L. Tran-Thanh, B. Wilder, E. Rice, P. Vayanos, M. Tambe, Utilizing housing resources for homeless youth through the lens of multiple multi-dimensional knapsacks, in: 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018, pp. 41–47.

[29] L. Blumrosen, N. isan, Combinatorial auctions, in: Algorithmic Game Theory, 2007, pp. 267–300.

[30] R. Myerson, Optimal auction design, Math. Oper. Res. 6 (1981) 58–73.

[31] T. Bahreini, H. Badri, D. Grosu, An envy-free auction mechanism for resource allocation in edge computing systems, in: 2018 IEEE/ACM Symposium on Edge Computing, 2018.

[32] https://github.com/alibaba/clusterdata/blob/v2018/cluster-trace-v2018/trace2018.md.

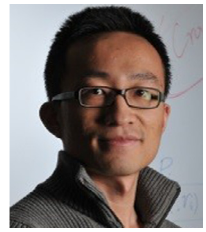
[33] http://www.modelingonlineauctions.com/datasets.

Weifeng Lu was born in 1979. He received the Ph.D. degree in Communication and Information System from Nanjing University of Posts and Telecommunications University in 2007. Since 2012, he has been working as a Postdoctoral Researcher in Information Science and Engineering of Southeast University. He is currently a associate professor in Jiangsu Key Laboratory of Big Data Security and Intelligent Processing at Nanjing University of Posts and Telecommunications. His research interest is in radio resource management in device-to-device networks and edge computing.

Weiduo Wu received the B.S. degree form the Nanjing University of Posts and Telecommunications, Nanjing, China in 2018. He is currently pursuing the M.E. degree in Jiangsu Key Laboratory of Big Data Security and Intelligent Processing at Nanjing University of Posts and Telecommunications, Nanjing, China. His main research interests is in edge computing.

Jia Xu received the M.S. degree in School of Information and Engineering from Yangzhou University, Jiangsu, China, in 2006 and the PhD. Degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in Jiangsu Key Laboratory of Big Data Security and Intelligent Processing at Nanjing University of Posts and Telecommunications. He was a visiting Scholar in the Department of Electrical Engineering & Computer Science at Colorado School of Mines from Nov. 2014 to May. 2015. His main research interests include crowdsourcing, edge computing and wireless sensor networks. Prof. Xu has served as the PC Co-Chair of SciSec 2019, Organizing Chair of ISKE 2017, TPC member of Globecom, ICC, MASS, ICNC, EDGE. He currently serves as the Publicity Co-Chair of SciSec 2021.

Pengcheng Zhao received the M.S. degree from Nanjing Forestry University and the B.S. degree in Chengxian College from Southeast University, Nanjing, China, in 2019 and in 2015, respectively. He is currently working toward the Ph.D. degree in Jiangsu Key Laboratory of Big Data Security and Intelligent Processing at Nanjing University of Posts and Telecommunications, Nanjing, China, since 2019. His research interests are mainly in the areas of the mobile crowdsensing, edge computing, and blockchain.



Dejun Yang received the B.S. degree in computer science from Peking University, Beijing, China, in 2007 and the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2013. Currently, he is an Associate Professor of computer science with Colorado School of Mines, Golden, CO, USA. His research interests include Internet of things, networking, and mobile sensing and computing with a focus on the application of game theory, optimization, algorithm design, and machine learning to resource allocation, security and privacy problems. Prof. Yang has served as the TPC Vice-Chair for Information Systems for IEEE International Conference on Computer Communications (INFOCOM) and currently serves an Associate Editor for the IEEE Internet of Things Journal (IoT-J). He has received the IEEE Communications Society William R. Bennett Prize in 2019 (best paper award for IEEE/ACM Transactions on Networking (TON) and IEEE Transactions on Network and Service Management in the previous three years), Best Paper Awards at IEEE Global Communications Conference (GLOBECOM) (2015), IEEE International Conference on Mobile Ad hoc and Sensor Systems (2011), and IEEE International Conference on Communications (ICC) (2011 and 2012), as well as a Best Paper Award Runner-up at IEEE International Conference on Network Protocols (ICNP) (2010).

Lijie Xu received the Ph.D. degree in computer science from Nanjing University, Nanjing, in 2014. He is currently an Associate Professor of the School of Computer Science at Nanjing University of Posts and Telecommunications, Nanjing. He was a research assistant in the Department of Computing at the Hong Kong Polytechnic University from November 2011 to May 2012. His research interests are mainly in the areas of wireless sensor networks, ad-hoc networks, mobile and distributed computing.