Tan HS, Li GP, Shen ZY *et al.* Edge Pricing Mechanisms with Selfish Users. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(1): 1–22 January 2024. DOI 10.1007/s11390-015-0000-0

Edge-Centric Pricing Mechanisms with Selfish Heterogeneous Users

Hai-Sheng Tan[1,2] (谈海生), *Member, CCF, ACM, Senior Member, IEEE*, Guo-Peng Li[1] (李国鹏), *Member, ACM*, Zi-Yu Shen[1] (沈子喻), Zi-He Wang[3,*] (王子贺), *Member, CCF*, Zhen-Hua Han [4] (韩震华), *Member, ACM*, Ming-Jun Xiao[1] (肖明军), *Member, CCF, ACM, IEE*E, Xiang-Yang Li[1,2] (李向阳), *Senior Member, CCF, Fellow, ACM, IEEE* and Guo-Liang Chen[1,2] (陈国良), *Fellow, CCF*

[1] *School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China*

[2] *USTC-Deqing Alpha Innovation Research Institute, Huzhou 313200, China*

[3] *Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China*

[4] *Microsoft Research Asia, Shanghai 200232, China*

E-mail: hstan@ustc.edu.cn; guopengli@mail.ustc.edu.cn; shenziyu@mail.ustc.edu.cn; wang.zihe@ruc.edu.cn; Zhenhua.Han@microsoft.com; xiaomj@ustc.edu.cn; xiangyangli@ustc.edu.cn; glchen@ustc.edu.cn

Received April 14, 2023; accepted January, 2024.

Abstract Through deploying computing resources close to users, edge computing is regarded as a promising complement to cloud computing to provide low-latency computational services. Meanwhile, edge platforms also play the role of competitors of the cloud platforms in a non-cooperative game, which sets prices for computational resources to attract users with different real-time requirements. In this paper, we propose the edge pricing game under competition (EPGC) and investigate the truthful pricing mechanisms of the edge platform with the objective of maximizing its revenue under three different settings. When all user information is available, the optimal mechanism (OM) can be achieved based on a knapsack problem oracle. With partial information, where users' resource demand is given but their preference information to the edge platform is private, we propose a random sampling mechanism (RSM) that achieves a constant approximation with probability approaching one. We also propose an efficient heuristic greedy mechanism, and we call it GM. Both mechanisms are truthful, GM is directly applicable, while RSM requires minor modifications (RSM^+) for deployment in the prior-free setting where all user information is private. Finally, extensive simulations are conducted on the Google cluster dataset. The results validate our theoretical analysis that RSM^+ works well in the market where edge resources are scarce, while GM performs better when the edge platform has a larger capacity constraint.

Keywords edge computing, mechanism design, network economics

Regular Paper

This work was supported in part by the National Key R&D Program of China under Grant No. 2021ZD0110400, the Key Program of the National Natural Science Foundation of China under Grant No. 62132009 and the Fundamental Research Funds for the Central Universities at China.

[*]Corresponding Author

A preliminary version of the paper was published in the Proceedings of CWSN 2020.

1 Introduction

The Internet of Things (IoT) and mobile computing have been developing rapidly and have drawn extensive attention from both academia and industry. For example, applications like voice assistants need to perform computation-intensive speech recognition while providing services to users. To overcome the limitations of computational resources and energy, these applications often offload such tasks to remote data centers. This computing paradigm is known as mobile cloud computing (MCC), which has enabled many convenient services. However, some emerging applications like augmented reality and autonomous driving require real-time video processing [1]. MCC's long propagation delay brought by the geographical distance between the data and the remote cloud data center becomes a major shortcoming. To mitigate this problem, edge computing has been proposed by deploying small-scale edge servers at the edge of the Internet, which is close to the users [2–4]. Edge computing can respond to real-time applications in a timely manner and is regarded as the key technology to achieve the 6G vision [5, 6].

To promote the development of edge computing, a suitable pricing mechanism is crucial for ensuring economic benefits. There have been many studies on pricing mechanisms in cloud computing [7–10]. Three widely adopted types are: 1) *Pay-as-you-go*: a user pays the cloud service provider (CSP) a fixed amount of money every period (e.g., several minutes or hours) based on computing requirements. CSPs often provide elastic computing services, which allow users to pay different amounts for Virtual Machines (VMs) with varying computing power as needed during different periods. 2) *Subscription*: a user selects a specific type of VM based on his demands and pays the CSP a subscription fee. During the subscription interval, computing resources (such as CPU, memory, and storage) are reserved for the subscriber. 3) *Auction*: a user sets a maximum payment (bid) per instance over a certain time period. The CSP performs an auction to determine the price at certain intervals, and the user's computing request will be served if the bid is higher than the price. However, there is limited research on pricing issues in edge computing scenarios, while the pricing mechanism of the edge platform is quite different from that of the cloud platform [11]. This paper is the extended version of our preliminary work [12]. In this version, we introduce new theoretical insights, offer a more comprehensive evaluation, and present an extensive survey of the literature.

Designing a pricing mechanism for the edge platform is challenging for the following reasons:

1) *Competition*. The edge platform may need to compete with the cloud platform. Let us consider a face recognition process that can be executed on both the cloud platform and the edge platform. The edge platform offers a better experience with lower response time and better privacy, while the cloud platform provides services at a lower price. The two platforms have a competitive relationship. Therefore, the pricing mechanism of the edge platform is constrained by the cloud platform, and a proper charging mechanism needs to be adopted at the edge platform to avoid users choosing the cloud platform instead.

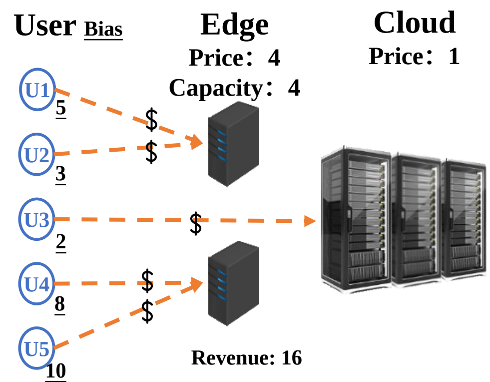
2) *Limited Resources*. The edge platform has relatively limited computational resources compared with the cloud platform, making it necessary to design the pricing mechanism to maximize revenue carefully. Besides, even when the optimal prices are given, the optimal allocation rule determining which tasks should be executed on the edge platform is a challenging problem.

3) *Heterogeneous Users*. The type and size of resource requirements by users are often heterogeneous. Additionally, different users have varying real-time requirements, resulting in different preferences when choosing between the cloud platform and the edge platform. Therefore, it is crucial to leverage this property to maximize revenue for the edge platform.

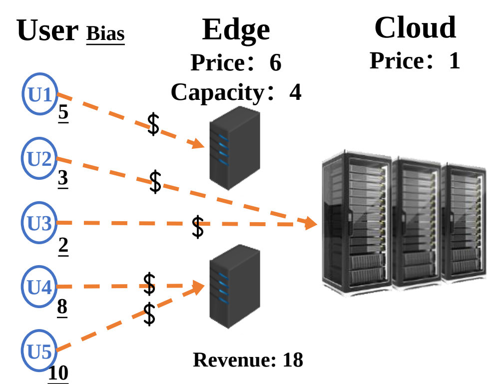
4) *Private Information*. The users' resource demands and preferences are typically private information that the edge platform cannot directly observe. This makes it challenging to design a pricing mechanism that accurately reflects each user's actual demands and preferences. The combination of the above characteristics makes it challenging to design the pricing mechanism for the edge platform.

In this paper, we propose the edge pricing game under competition (EPGC) and investigate the pricing mechanisms of the edge platform with the objective of maximizing its revenue. Each user has varying computation resource requirements. The edge platform will decide the pricing of resources, and the users will purchase resources at the cloud platform or the edge platform. To fit the actual scenario, we consider the following general assumptions: 1) Multi-dimensional computational resources: typically, three kinds of resources are considered: CPU, memory, and storage. 2) Dynamic VM packing: in each time slot, the platform can dynamically pack VMs based on the computational resource the users request. 3) Indivisible computational resources: a user's required computational resources should all be allocated at the cloud platform or the edge platform. If a user's requirement can be allocated at both platforms, we will divide the case into two different users. 4) Bias of the edge platform: to model the personal preference to the edge platform of each user, i.e., the lower latency at the edge platform compared with the cloud platform, we define bias as the preference value of each user. If the total charge of the edge platform is higher than that of the cloud platform by no more than the bias, the user will tend to choose the edge platform. In order to maximize the revenue with limited resources at the edge platform, we should carefully set the price at the edge platform to compete with the cloud platform. Here is a simple example illustrating that better pricing at the edge platform can get more revenue with fewer resources.

Example 1 (Motivating Example)**.** Fig. 1 illustrates an example with resources of a single dimension to show the importance of designing an appropriate pricing mechanism. We assume each user requires 1 unit of computational resource, and the capacity of the edge platform is 4.



(a)



(b)

Fig.1. Motivating example: a single dimension of resources. (a) The price of the edge platform is 4. (b) The price of the edge platform is 6.

There are five users (U1, U2, U3, U4 and U5) whose bias values are listed in the bottom right corner. When the price of the edge platform is 4 and the price of the cloud platform is 1, the price difference is 3. Under this situation, U1, U2, U4, and U5 will choose the edge platform because they have a bias greater than or equal to 3, while U3 will choose the cloud platform. The revenue of the edge platform and the cloud platform would be 16 and 1, respectively. If the edge platform changes its price from 4 to 6, then U2 will change its choice to the cloud platform, and the revenue of the edge platform and the cloud platform would be 18 and 2, respectively.

To the best of our knowledge, this is the first work to study the pricing mechanism for edge computing while considering the cloud platform's competition. Our contributions can be summarized as follows:

- We first propose a competition model EPGC between the cloud platform and the edge platform. Both platforms set multiple prices to attract users and maximize revenue (Section 2). We show that in the competition game, there does not always exist a Nash equilibrium (Section 3).
- We explore the pricing mechanism for the edge platform in comparison to the cloud platform, considering three scenarios based on the extent of user information available to the platforms: all, partial, and none. Specifically, when all information is accessible, the optimal mechanism (OM) can be achieved (Section 4). When partial information is accessible, we propose a heuristic greedy mechanism (GM) and a random sampling mechanism (RSM)(Section 5). Moreover, both GM and RSM can be extended to cases where no user information is available, RSM$^+$ is the modified version of RSM (Section 6).
- We evaluate OM, GM, and RSM$^+$ by extensive simulations based on the data-trace from the Google cluster [13]. Simulation results show that our mechanisms perform consistently well to maximize the revenue with different settings of various parameters. Most notably, in experiments conducted using Google trace, RSM$^+$ can achieve near-optimal revenue for the edge platform. (Section 7).

2 System and Problem Definition

Network Model. We consider a time-slotted network that each computing task will finish in a single slot, which is reasonable in the edge computing scenario. And without loss of generality, we can focus on a single time slot. There are two computing service providers: the cloud platform and the edge platform. There are multiple dimensions of computational resources, and we denote the number of dimensions as τ. Here, we consider $\tau = 3$, i.e., there are three dimensions of computational resources: CPU, memory and storage, denoted by $g = 1, 2, 3$, respectively. Regarding the available amount of computational resources, the edge platform has limited resources, while the cloud platform has unlimited resources. Specifically, we denote the available resources of different types at the edge platform as C_e^g where $g = 1, 2, 3$. Important notations are listed in Table 1.

Table 1. List of Notations

Notation	Description
τ	Dimensions of computational resources
C_e^g	Available resources of different types at the edge platform, $g = 1, 2, .., \tau$
p_e^g	Pricing of the resource g at the edge platform, $g = 1, 2, .., \tau$
p_c^g	Pricing of the resource g at the cloud platform, $g = 1, 2, .., \tau$
u_e	Revenue of the edge platform
u_c	Revenue of the cloud platform
d_i^g	User i's required of resources g, $g = 1, 2, .., \tau$
v_i	User i's preference of the edge platform

The Pricing Model. The relationship between the cloud platform and the edge platform is competitive, with both platforms striving to maximize revenue by renting computational resources to users for task execution. In practice, the scale of the cloud platform is typically much larger than that of the edge platform. We assume the cloud platform sets prices p_c^g for using a unit of computational resources of type g, $g = 1, 2, 3$ at first. Then, the edge platform determines the pricing of its three different resources p_e^g to maximize its revenue in response to the cloud platform's pricing. Note that if the total resource demand of the edge platform exceeds its capacity, the edge platform will choose and rent computational resources to a subset of the users choosing the edge platform, and the other users' requests will be offloaded to the cloud platform. The revenue of the platform is the total payment from the users minus the cost of providing computing resources. Here, we assume the cost is zero for simplicity. All the results can be easily extended to the model where the cost is not zero. The revenue of the cloud platform and the edge platform are denoted as u_c and u_e, respectively.

User Model. There are n users, each with a computation task to be executed either at the cloud platform or the edge platform. User i's required resources are denoted as d_i^g, and the cloud platform or the edge platform will pack the resources as a VM to execute the user's request. Users choose to purchase resources from one platform based on two price profiles p_e^g and p_c^g. Since the edge platform can complete computation tasks with lower latency than the cloud platform, users prefer the edge platform. This preference is modeled as a bias v_i for user i, such that if the edge platform charges user i no more than v_i compared to the cloud platform, the user will choose the edge platform. Specifically, user i is represented by vector $(d_i^1, ..., d_i^\tau, v_i)$, and chooses the edge platform if Eq. (1) holds.

$$\sum_{g=1}^{\tau} d_i^g p_e^g \leq \sum_{g=1}^{\tau} d_i^g p_c^g + v_i. \tag{1}$$

In this paper, the user's utility is defined as the user's bias minus the payment he makes to the edge platform, and when the user is not selected by the edge platform, his utility is 0. The welfare is defined as the sum of biases of users chosen by the edge platform. In other words, welfare is the social improvement due to the existence of the edge platform. While the revenue is platform-oriented, the revenue of the edge platform is defined in Eq. (2), where E is the set of users whose computation tasks are executed at the edge platform. Even user i is willing to choose the edge platform given the prices, it is possible that user i is not chosen by the edge platform due to the capacity constraint.

$$u_e = \sum_{i \in E} \sum_{g=1}^{\tau} p_e^g d_i^g. \tag{2}$$

Problem Formulation. Based on the settings above, we formulate the edge pricing game under competition (EPGC) [12].

Game 1 (EPGC)**.** *The edge platform aims to maximize its utility. Given the prices on the cloud platform are $\{p_c^g\}$, the edge platform then decides prices $\{p_e^g\}$. The users will choose the cloud platform or the edge platform based on their resource demands and biases, $\{(d_i^1, d_i^2, d_i^3, v_i)\}$. Next, the edge platform chooses a set of users from the candidates subject to the resource constraint $\{C_e^g\}$.*

In the following sections, for Nash equilibrium analysis (Section 3), we mainly focus on the pricing strategies. When designing mechanisms, we take both the pricing mechanisms and the candidate selection strategies into consideration.

3 Nash Equilibrium Analysis

In this section, we prove that there does not always exist a Nash equilibrium in this game. A Nash equilibrium of a game is that none of the players has the incentive to deviate from his current strategy to achieve higher utility. We claim that there is no Nash equilibrium when there is only one type of resource, i.e., $\tau = 1$. It can be easily generalized to the case for $\tau > 1$.

In general, we allow for a continuous pricing space in $\mathbb{R}$. Suppose there exists a user who has no preference between choosing the cloud platform or the edge platform, implying that Eq. (3) holds true.

$$\sum_{g=1}^{\tau} d_i^g p_e^g = \sum_{g=1}^{\tau} d_i^g p_c^g + v_i. \tag{3}$$

The cloud platform can improve its revenue by slightly reducing the price, attracting more users while sacrificing only a small portion of its price, causing both platforms to decrease their prices. Therefore, a Nash equilibrium may not exist. However, if we restrict the pricing space to be discrete, such as integers, platforms cannot adjust prices continuously. Despite this restriction, the non-existence of a Nash equilibrium still holds in this setting.

Theorem 1. *There does not always exist a Nash equilibrium.*

Proof. We present a hard instance to show the non-existence of Nash equilibrium. We consider there is one type of resource, i.e., $\tau = 1$. There are three users: $(d_1 = 10, v_1 = 3)$, $(d_2 = 3, v_2 = 2)$, and $(d_3 = 6, v_3 = 1)$ and the capacity on the edge platform is $C_e^1 = 19$. Since $C_e^1 = 19$ is large enough to contain all users' tasks, the edge platform will choose any user who can afford the prices.

Let p_e^1 and p_c^1 denote the prices set by the edge platform and the cloud platform, respectively. For any combination of the edge platform prices and the cloud platform prices, four cases are listed in Table 2.

Table 2. A Hard Instance to Show the Non-existence of Nash Equilibrium

Case	Price Relationship	Edge Revenue	Cloud Revenue
1	$p_e^1 > p_c^1 + 3$	0	$(10 + 3 + 6) \times p_c^1$
2	$p_e^1 = p_c^1 + 3$	$10 \times p_e^1$	$(3 + 6) \times p_c^1$
3	$p_e^1 = p_c^1 + 2$	$(10 + 3) \times p_e^1$	$6 \times p_c^1$
4	$0 \leq p_e^1 \leq p_c^1 + 1$	$(10 + 3 + 6) \times p_e^1$	0

- For case 1, the edge platform price p_e^1 will decrease to a positive value so that the edge platform will have a positive revenue.

- For case 2, if $p_c^1 \leq 1$, the cloud platform price will increase its price by 1 to improve its revenue, while if $p_c^1 > 1$, the edge platform price will decrease by 1 to get higher revenue.

- For case 3, if $p_e^1 \leq 3$, the edge platform price will increase by 1 to improve its revenue. Otherwise, the cloud platform will adjust its price to the situation in case 2 to get higher revenue.

- For case 4, if $p_e^1 \leq 2$, the edge platform will increase its price, otherwise, the cloud platform will adjust its price to the situation in case 3 to gain higher revenue.

For all combinations of prices, either the cloud platform or the edge platform will change its price for higher revenue. It is easy to generalize this proof to multiple resource types and fractional prices. Therefore, there is no Nash equilibrium. This completes the proof. □

Since there does not always exist a Nash equilibrium, we will focus on the pricing strategy. In practical scenarios, the cloud platform has a much larger scale than the edge platform, and thus, the edge platform's pricing strategy has a negligible impact on the revenue of the cloud platform. Therefore, we focus on the edge platform's pricing mechanism. Specifically, we investigate the optimal pricing mechanism under the assumption that all information is public. Furthermore, we propose truthful mechanisms for the scenarios of partial prior information and no prior information.

4 All User Information

In this section, we will consider mechanism design under all information, where every user's type, including his resource demand and bias, is public information. As mentioned, we focus on the pricing mechanism of the edge platform. Although a Nash equilibrium might not exist, the edge platform always has an optimal revenue mechanism for each set of users with different biases and demands. In a mechanism, we need to specify the prices and the set of users chosen to be served by the edge platform due to the limited computational resources. We focus on designing a "fair" mechanism in which we do not set discriminate prices for different users, and the payment is proportional to how many resources a user rents on the edge platform. Theorem 2 proves that the difficulty of designing the optimal mechanism is NP-hard by reducing it to the subset sum problem [12].

Theorem 2. *Designing the optimal mechanism for the edge platform, even with a fixed cloud platform pricing strategy, is NP-hard.*

Proof. The proof is based on a reduction to the subset sum problem. Given a subset sum instance that the set of numbers is $\{a_1, a_2, ..., a_n\}$, we ask whether there is a subset such that the sum of the set is s. We construct the mechanism design problem with n users with type $(a_i, 0, 0, a_i), i = 1, ..., n$, and the capacity of resources on the edge platform is $(s, 0, 0)$. Obviously, the optimal prices are $(1, 0, 0)$. All users' values per unit resource are 1, which are equal. A mechanism that achieves revenue s exists if and only if there is a solution for the subset sum problem. □

We do not set negative prices for resources. The following lemma gives a characterization of the optimal prices [12].

Lemma 1. *If prices for $k \in [0, ..., \tau - 1]$ type resources are zero in the optimal mechanism, then there are at least $\tau - k$ users who exhibit no preference between the two platforms.*

The proof is omitted, and the idea is as follows. We first normalize the resource requirements proportionally, representing the resource cost per unit bias, and then each user can be represented by a point in $(\tau - k)$-dimensional space. $\tau - k$ users can determine a hyperplane that divides the space into three parts, representing that choosing the edge platform has higher/same/lower utility than the cloud platform, respectively. If there are less than $\tau - k$ users who exhibit no preference between the two platforms, we can always adjust the prices and increase the edge's revenue while the users make the same choice. With Lemma 1, we can enumerate all possibilities where users exhibit no preference between choosing either of the two platforms. By solving an equation set of the users' utility equivalence, we can determine the pricing and calculate the corresponding revenue. And the pricing with the highest revenue is the optimal pricing.

Mechanism 1 describes the optimal mechanism (OM) of the edge platform. It first enumerates a set of users who exhibit no preference between the two platforms (i.e., S). `OptimalAssignment`(S) takes S as the input to find the optimal prices to get the maximum revenue (Line 3–Line 9). For any combinations of three users/two users/one user (Line 11–Line 16), we can find one/three/three sets of the edge platform price and the cloud platform price plans such that the equities of in Eq. (1) holds for the chosen users, and there are zero/one/two types of resources' whose price is 0 (Line 4). Here, Eq. (1) represents the conditions that users must satisfy when selecting the edge platform. Then, with each fixed price plan, we can use a three-dimensional knapsack oracle to find the optimal set of users to be placed on the edge platform (Line 5). The knapsack solver returns the set of users chosen and the total revenue. In this way, we can enumerate all the potential optimal prices, which have $\binom{n}{3} + 3 \times \binom{n}{2} + 3 \times \binom{n}{1}$ cases in total. Mechanism 1 needs to find the optimal knapsack solution for $O(n^3)$ times. The overall time complexity is $O(n^3 \cdot KS(n))$, where $KS(n)$ is the time complexity of finding the optimal knapsack solution which is pseudo-polynomial.

Mechanism 1: Optimal Mechanism with All Information (OM)

Input
$(d_i^g, v_i), p_c^g, C_e^g, g = 1, ..., \tau, i = 1, ..., n;$
$u_e = 0, A = \emptyset$
Function `OptimalAssignment(`S`)`:
  for all sets of $\{p_e^g\}$ such that $|\{p_e^g : p_e^g > 0 \ \forall \ g\}| = 3 - |S|$ and $\sum_{g=1}^{\tau} d_i^g p_e^g - (\sum_{g=1}^{\tau} d_i^g p_c^g + v_i) = 0 \ (\forall \ i \in S)$ **do**
    $(A, u'_e) =$`OptimalKnapsack`$(\{(\{d_i^g\}, v_i), i \in S\}, \{C_e^g\})$
    if $u'_e > u_e$ **then**
      $u_e = u'_e$
      $\{p_e'^g\} = \{p_e^g\}$
  return $A, u_e, \{p_e'^g\}$
$u_{opt} = 0$
for all combinations S of three users i, j, k; two users i, j and one user i **do**
  $(A, u_e, \{p_e'^g\}) =$ `OptimalAssignment(`S`)`
  if $u_{opt} < u_e$ **then**
    $u_{opt} = u_e$
    $\{p_e^{opt,g}\} = \{p_e'^g\}$
    $A_{opt} = A$
return $A_{opt}, \{p_e^{opt,g}\}$

5 Partial User Information

In this section, we consider the scenario that a user's demand is public, while his bias information is private. We aim to design a competitive mechanism, meaning that it can yield a constant factor of the optimal revenue in the full information setting. To achieve this, we assume any user can only contribute a small fraction to the optimal revenue.

Following the principle of revelation [14], we only need to focus on truthful mechanisms. In this paper, we adopt the concept of ex-post truthfulness. We define a mechanism to be ex-post truthful if a user's best strategy is revealing type information truthfully no matter what other users report. An ex-post mechanism makes it easier for a user to make a decision since it does not need to consider the actions of other users. An ex-post truthful mechanism is characterized by the property that, when the edge platform selects a user, his bid is independent of his payment, with the latter depending solely on other users' bids. Thus a user's payment only depends on other users' bids.

It is clear that designing a revenue maximization mechanism is impossible in the prior free setting. For instance, there exists a user with a significant bias towards the edge platform, while the biases of all other users are comparatively negligible. In such a case, any ex-post truthful mechanism cannot always achieve optimal revenue, as the payment from the user with a high bias is fixed. Finally, we consider the design of the ex-post truthful mechanism. We first solicit users' bids, and users will truthfully report their biases. Then we set prices for three resources and determine who would be chosen by the edge platform. There are two basic ideas for setting prices: the first is to set the equilibrium prices when supply equals the demand, and the second is to set more reasonable prices by learning the distribution of the bias.

5.1 Greedy Mechanism

Following the first idea, we propose a greedy mechanism (GM). We first normalize capacities for the three resources to be the same. As shown in Mechanism 2, GM sorts all users in the decreasing order of bias per unit resource. Users are chosen sequentially until the demand for some resources exceeds the corresponding capacity of the edge platform. Then prices are set to the bias per unit resource of the user at whom the process stops. User i's per unit bias is defined as $\frac{v_i}{d_i^1 + d_i^2 + d_i^3}$ [12].

GM takes the demand as input and collects users' private information. Finally, it outputs the set of users chosen by the edge platform and corresponding prices for different types of resources.

GM works well when the demands for the three resources are not heavily correlated, resulting in a balanced resource consumption outcome. However, in cases where users with high per unit bias (q_i) only demand a single resource, GM may perform poorly. For instance, let us consider Example 2.

Mechanism 2: Greedy Mechanism (GM)

Input $d_i^g, p_c^g, C_e^g, g = 1, ..., \tau, i = 1, ..., n$;
All users report their biases $v_i, i = 1, ..., n$
$S = \emptyset$
for all users i **do**
  $q_i = \frac{v_i}{d_i^1 + d_i^2 + d_i^3}$.
Sort q_i such that $q_{m(1)} \geq q_{m(2)} \geq ... \geq q_{m(n)}$.
for $k = 1, ..., n-1$ **do**
  if $\sum_{i \in S} d_i^g + d_{m(k)}^g \leq C_e^g$ *for* $g = 1, 2, 3$. **then**
    $S = S \cup \{m(k)\}$
  else
    break
return S,$\{p_c^g + q_{m(k)}\}$

Example 2. *We assume that there are 7 users, and* $p_c^g = 0, C_e^g = 3$, *for* $g = 1, 2, 3$. *The first three only need the first type of resources. The other four users need all three types. Particularly, for users* $i = 1, 2, 3$, $(d_i^1, d_i^2, d_i^3, v_i) = (1, 0, 0, 1)$, *and for* $i = 4, 5, 6, 7$, $(d_i^1, d_i^2, d_i^3, v_i) = (1, 1, 1, 2.9)$.

The edge platform will choose users $1, 2, 3$ and set prices $(2.9/3, 2.9/3, 2.9/3)$ applying GM, and the revenue is 2.9. Since we have three types of resources and the users will only provide a bias in a single dimension so that we can have different definitions of the bias per unit. One possible modification is that user i's per unit bias is defined to be $\frac{v_i}{\max\{d_i^1, d_i^2, d_i^3\}}$. The edge platform will choose users $4, 5, 6$ and set prices $(2.9, 2.9, 2.9)$ applying the modified GM, and the revenue would be 26.1.

Theorem 3 shows that when GM sets the prices of the three resources to be the same (special case), and the revenue is at most $1/n$ of the optimal revenue, where n is the number of users [12]. The idea is to construct an example that can achieve a $1/n$ fraction of the optimal revenue at most. The detailed proof is presented in Appendix A1[1].

Theorem 3. *GM sets the same prices for three resources can only guarantee* $1/n$ *fraction of the optimal revenue, where* n *is the number of users.*

Since GM sets the same prices for three resources, we claim that:

Corollary 1. *GM cannot guarantee more than* $1/n$ *fraction of the optimal revenue.*

GM has a good welfare guarantee. As described in the previous sections, welfare is defined as the sum of biases of users chosen by the edge platform. In other words, welfare is the social improvement due to the existence of the edge platform.

Theorem 4. *Assume every user has less than* $1/\beta$ *fraction of the largest welfare that could be achieved. GM guarantees* $1/3 - 1/\beta$ *fraction of the largest welfare.*

The idea is to analyze the user not chosen by the edge platform with the highest per unit bias. Detailed proof is presented in Appendix A2.

5.2 Random Sampling Mechanism

Mechanism 3 [12] presents the random sampling mechanism (RSM), which involves the following steps. First, after all users report their biases, a subset of users is randomly sampled. Second, the biases of the sampled users are learned, and optimal prices are computed based on the samples. Third, the computed prices are applied to the remaining users. Finally, the mechanism solves a knapsack problem to allocate resources among the selected users.

[1] Due to the length constraints, please visit the link to access the appendix. https://gitee.com/USTC_cs/pricing, Jan. 2024.

Mechanism 3: Random Sampling Mechanism (RSM)

Input $d_i^g, p_c^g, C_e^g, g = 1, ..., \tau, i = 1, ..., n$;
All users report their bias $v_i, i = 1, ..., n$
$S = \emptyset$
for all users i **do**
    With probability $\frac{1}{2}$, $S = S \cup \{i\}$
$(S_1, \{p_e^g\}) =$ Mechanism $1(\{(\{d_i^g\}, v_i), i \in S\}, \{C_e^g\})$
$T = \{i \in N \backslash S | v_i \geq \sum_{g=1,...,\tau} p_e^g * d_i^g\}$
$(A, u_e) =$`Knapsack`$(\{(\{d_i^g\}, v_i), i \in T\}, \{C_e^g\})$
return $A, \{p_e^g\}$

In Theorem 5, by analyzing the strategy of a user i, we can find out that the dominant strategy is to report his type truthfully.

Theorem 5. *RSM is ex-post truthful.*

Proof. When user i is chosen as a sample, he has no incentive to misreport since its utility is doomed to be zero. When user i is not chosen as a sample, the price is fixed already, and what bias he reports only determines whether he would be in set T. Supposing user i is in T, whether he will be chosen in A is already determined by his demand, which is public information. If user i misreports and is included in set A, then he will only get negative utility since his bias cannot afford the payment. On the other hand, if user i misreports and is not included in set T, he will have zero utility. If user i misreports and is not a member in T as a result, then he will have zero utility. In summary, misreporting will not improve a user's utility in any case. Therefore, RSM is ex-post truthful. □

Next, we show a lemma (Lemma 2) that would be used in the proof of RSM (Mechanism 3) is competitive [12].

Lemma 2. *For any i, x_i is a random variable equals 0 and h_i with equal probability and $E[x_i] = h_i/2$. Let $\bar{h} = \max_i\{h_i\}$ and we assume $\bar{h} \leq \frac{1}{\beta}\sum_i x_i$. Then we have $\Pr(\sum_i x_i \notin [\frac{\sum_i h_i}{3}, \frac{2\sum_i h_i}{3}]) \leq 2\exp(-\frac{\beta}{18})$.*

The proof is in Appendix A3. We run RSM, and for each user, we pick him with half probability. We denote the sample set of users by S and the n users by N. We denote the optimal price and revenue for n users in the all information setting by $\{p_0^g\}$ and R_0. We assume that any single user cannot contribute more than $\frac{1}{\beta}$ fraction of the revenue R_0.

Theorem 6. *With probability at least $1 - n^3\exp(-\frac{\beta}{54}) - 2\exp(-\frac{\beta}{18})$, we can achieve at least $R_N/9$ revenue using the sampling mechanism.*

Theorem 6 proves that, for RSM, the edge platform obtains a constant approximation of the optimal revenue with a probability close to 1. The general idea of the proof is to bound the revenue of various subsets of users and utilize Lemma 2 to demonstrate that the deviation from the expected revenue is minimal. A more detailed proof can be found in Appendix A4.

6 No User Information

In this section, we consider the scenario where a user's resource demand and bias are both private information. As discussed in Section 5, designing a mechanism that maximizes revenue is impossible in the prior-free setting, and users will truthfully report their biases. If a user misreports his information [15–17], he would only be incentivized to report a higher demand to receive a greater allocation potentially. Otherwise, even if the user is selected by the edge platform, it would not be able to complete the task due to insufficient resources. We still focus on the truthful mechanism and consider if the mechanism introduced in the previous section can be used in this setting.

Theorem 7. *GM truthful under the no user information setting.*

Theorem 7 demonstrates that GM is truthful in a scenario where users have no information. The proof aims to establish that regardless of the information reported by other users, a user's dominant strategy is to reveal his true type, which is achieved by analyzing user i's ranking in Mechanism 2. Detailed proof can be found in Appendix A5.

Mechanism 4: Modified Random Sampling (RSM$^+$)

```
Input d_i^g, p_c^g, C_e^g, g = 1, ..., τ, i = 1, ..., n;
All users report their types {(d_i^g, v_i)}
S = ∅
for all users i do
    S = S ∪ {i} with probability 1/2
(S_1, {p_e^g}) = Mechanism 1({({d_i^g}, v_i), i ∈ S}, {C_e^g})
T = {i ∈ N\S | v_i ≥ Σ_{g=1,...,τ} p_e^g * d_i^g}
A = ∅
for i ∈ T in random order do
    if d_i^g + Σ_{j∈A} d_j^g ≤ C_e^g for g = 1, 2, 3 then
        A = A ∪ {i}
return A, {p_e^g}
```

We also observe that the RSM is not truthful. Here is an example: after sampling, the mechanism learns the optimal prices on the sampling users are $(1, 1, 1)$. The capacity of the edge platform is $(4, 4, 4)$. There are two users left after sampling: user 1's type is $(2, 2, 2, 20)$ and user 2's type is $(3, 3, 3, 10)$. Both users 1 and 2 can afford the prices. In RSM, user 2 will be chosen by the edge platform since user 2 has a larger demand. As a result, user 1's utility is zero. This mechanism is not truthful since user 1 has an incentive to report $(4, 4, 4, 20)$, which leads to a higher utility $20 - 4 - 4 - 4 = 8$.

The reason why RSM fails to be truthful is that users are chosen using a knapsack algorithm, which depends on users' demand. We propose a modified mechanism (Mechanism 4, RSM$^+$), where we first arrange the users in uniformly random order and then accept users sequentially if they fit on the edge platform.

Theorem 8. *RSM$^+$ is truthful.*

Proof. The general idea is that a user has no incentive to move from T to $N \backslash T$ and vice versa. If a user is already a member of T, reporting a greater demand will only cause a larger payment. Therefore, RSM$^+$ is truthful. □

7 Evaluation

In this section, we evaluate OM, GM, and RSM$^+$ by extensive simulations on a real-world datatrace.

7.1 Experiment Settings

By default, we use the data set of Google cluster [13] to obtain the resource requirement of tasks. We divide the tasks by their release time into different time slots. We choose OM of the edge platform as a baseline, and need to solve the three-dimensional knapsack problem $O(n^3)$ times for a time slot. Thus, we choose a relatively small scale data of 150 random tasks of a minute to evaluate the performance by default. As for the pricing of the cloud platform, we use the pricing of Google cloud[2].

For ease of representation, we normalize the price to 280 per CPU core, 35 per GB of memory, and 1 per GB of disk, and the capacity of the edge platform of all resources is normalized to 1 by default. There is no existing data on the bias. Therefore, the bias of each task is generated randomly from 1 to 10 by default. In the following experiments, the default number of users is 150.

7.2 Mechanism and Metrics

Under the experiment settings above, we implement and compare OM, GM, RSM$^+$, the dynamic pricing based double auction ([18], represented by

[2] https://cloud.google.com/compute/all-pricing, Aug. 2023.

"DPDA") and GM for edge resource allocation problem ([19], represented by "G-ERAP"). The main idea of DPDA and G-ERAP will be introduced in Section 8. We also compare these mechanisms with fixed pricing, which is widely adopted by current service providers.

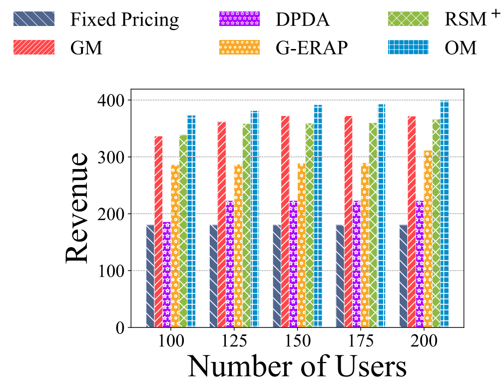
Note that we can set different sampling rates for RSM$^+$, and the sampling rate is set to $1/2$ by default. We evaluate different mechanisms by comparing the corresponding revenue of the edge platform, and the revenue is the sum of all users' normalized payments whose computation tasks are executed at the edge platform.

7.3 Simulation Results

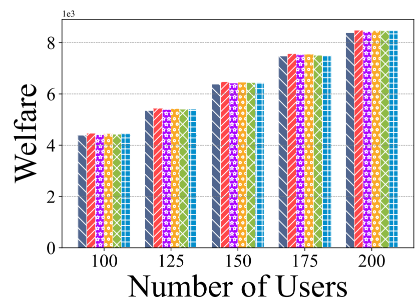
Here, we present the simulation results on the performance of the pricing mechanisms on the impact of the number of users, the impact of the distribution of users' resource demands, the capacity of the edge platform, the biases of users, and the sampling rate.

7.3.1 Impact of the Number of Users

Fig. 2 illustrates the revenue of the edge platform and the social welfare with the number of users from 100 to 200. As Fig. 2(a) illustrates, the revenue of the fixed pricing mechanism remains constant regardless of the number of users, while the revenue of OM increases slightly and then stabilizes. RSM$^+$ and GM achieve a higher revenue than the DPDA and G-ERAP mechanisms. Specifically, RSM$^+$ improves the revenue by at least 60.65% over DPDA and by at least 17.7% over G-ERAP. GM improves the revenue by at least 61.94% over DPDA and by at least 17.9% over G-ERAP. It should be noted that when there are 150 or 175 users, GM performs better than RSM$^+$. This is because RSM$^+$ excludes half of all the users, which leads to underutilization of the computational resources at the edge platform. This drawback can be overcome by adjusting the sampling rate, which we will discuss later. As for social welfare, it is mainly affected by the number of users. GM achieves the highest social welfare among all mechanisms, followed by OM and RSM$^+$.



(a)



(b)

Fig.2. Impact of the number of users. (a) Revenue. (b) Social Welfare.

7.3.2 Impact of the Distribution of Users' Resource Demands

The users' resource demand data used in Subsection 7.3.1 are from Google cluster. To evaluate the performance of the mechanisms proposed in this paper under different distributions of users' resource demands, we generate users' resource demands following the common distributions used in edge computing and cloud computing, namely normal distribution [20], log-normal distribution [21], and Pareto distribution [22, 23], and conduct experiments.

The experiment results are shown in Fig. 3. Fig. 3(a) shows that when users' resource demands follow a normal distribution, the revenue of OM and RSM$^+$ increases with the number of users, while GM's revenue changes slightly. However, RSM$^+$

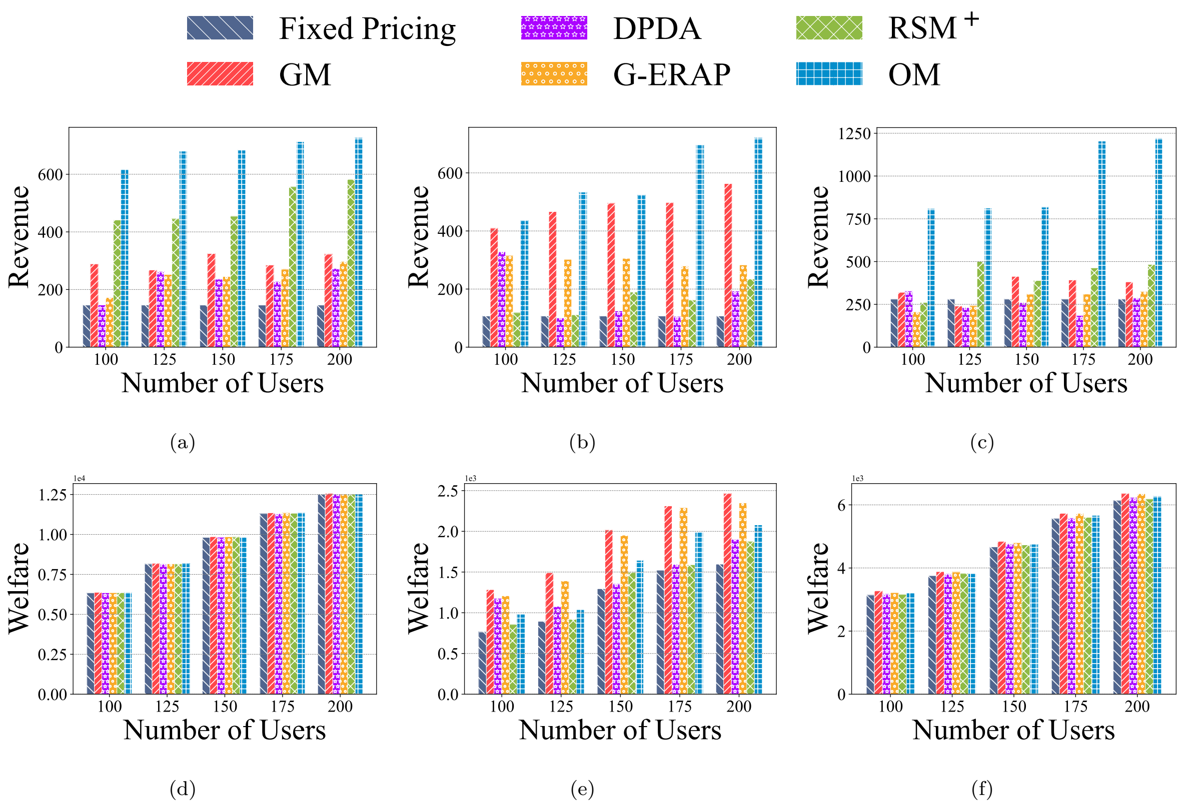


Fig.3. Impact of the distribution of the users' resource demand: normal, log-normal and Pareto distributions. (a) Revenue - normal. (b) Revenue - log-normal. (c) Revenue - Pareto. (d) Social welfare - normal. (e) Social welfare - log-normal. (f) Social welfare - Pareto.

and GM always outperform the DPDA and G-ERAP mechanisms in terms of revenue. For social welfare, there is no significant difference among different mechanisms under different numbers of users, as Fig. 3(d) illustrates. When users' resource demands follow a log-normal distribution, due to the significant differences in users' resource demands, RSM$^+$'s performance is worse, but GM still performs better than the DPDA and G-ERAP mechanisms, as shown in Fig. 3(b) and Fig. 3(e). When users' resource demands follow a Pareto distribution, as Fig. 3(c) illustrates, RSM$^+$ and GM perform better than the DPDA and G-ERAP mechanisms. In summary, when the distribution of users' resource demands changes, the performance of mechanisms changes accordingly, especially for RSM$^+$, but our proposed algorithm can achieve better performance than other mechanisms, and obtain similar or higher social welfare than OM.

7.3.3 *Impact of the Capacity of the Edge Platform*

Since we have performed normalization on the trace data and the price, we can assume the capacity of three resources increases or decreases by the same value with the number of machines at the edge platform increasing or decreasing. Fig. 4 demonstrates the revenue of the edge platform with the capacity of three resources from 0.6 to 1.4. As we can see, more computational resources lead to higher revenue. The revenue of all the mechanisms increases as the capacity of the edge platform increases. As for social welfare, the result is similar.

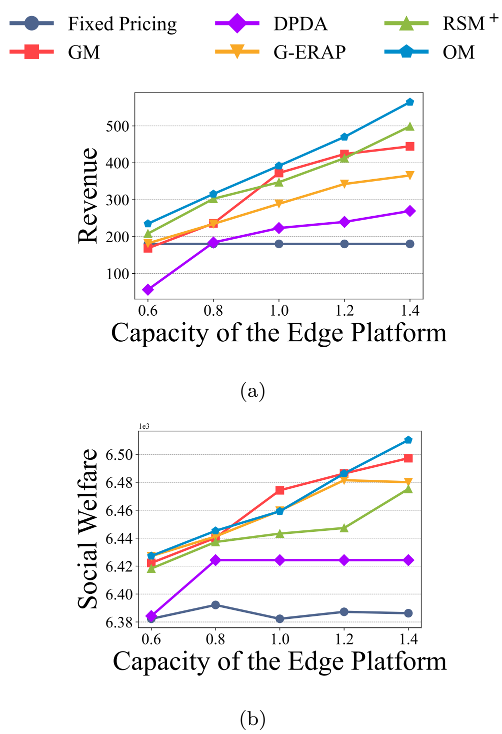


Fig.4. Impact of the capacity of the edge platform. (a) Revenue. (b) Social Welfare.

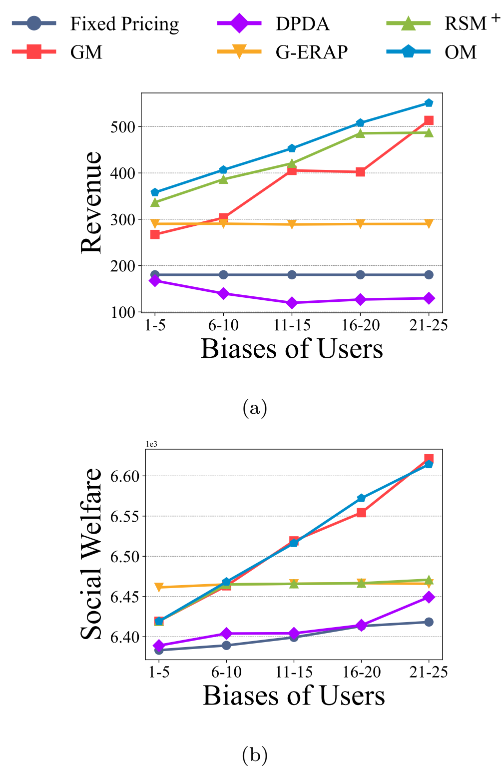


Fig.5. Impact of biases of users. (a) Revenue. (b) Social Welfare.

7.3.4 Impact of Biases of Users

Fig. 5 demonstrates the revenue of the edge platform of different ranges of biases of users. We simulate five groups of users, having biases of 1 to 5, 6 to 10, 11 to 15, 16 to 20 and 21 to 25, respectively. The simulation result is quite straightforward with higher bias, the users are willing to pay more for executing tasks at the edge platform. This leads to higher pricing for a unit of computational resource, and therefore the revenue becomes higher. As for social welfare, the higher bias also results in higher social welfare.

In Fig. 5, we investigate the impact of different ranges of users' biases on the performance of the mechanisms. Users' biases are generated randomly within specified ranges. Subsequently, experiments are conducted considering three distributions of users' biases: uniform, normal, and exponential [10]. The biases range from 1 to 10. Fig. 6 shows the results.

As shown in Fig. 6(a), when users' biases follow a uniform distribution, GM, DPDA and G-ERAP mechanisms' revenue does not change significantly as the number of users increases, because these three mechanisms are all based on the greedy principle of users' biases. For RSM$^+$, when users' biases follow a normal distribution or an exponential distribution, RSM$^+$'s revenue does not increase continuously as the number of users increases as shown in Fig. 6(b) and Fig. 6(c). Additionally, the experiment results show that different mechanisms achieve similar social welfare under different distributions of users' biases. Overall, this indicates that the distribution of users' biases affects the performance of RSM$^+$, which is reasonable because the core of RSM$^+$ is to learn users' biases.

7.3.5 Impact of Sampling Rate

We investigate the revenue of the edge platform with RSM$^+$ under different sampling rates. The sampled users' information is used to decide the pric-

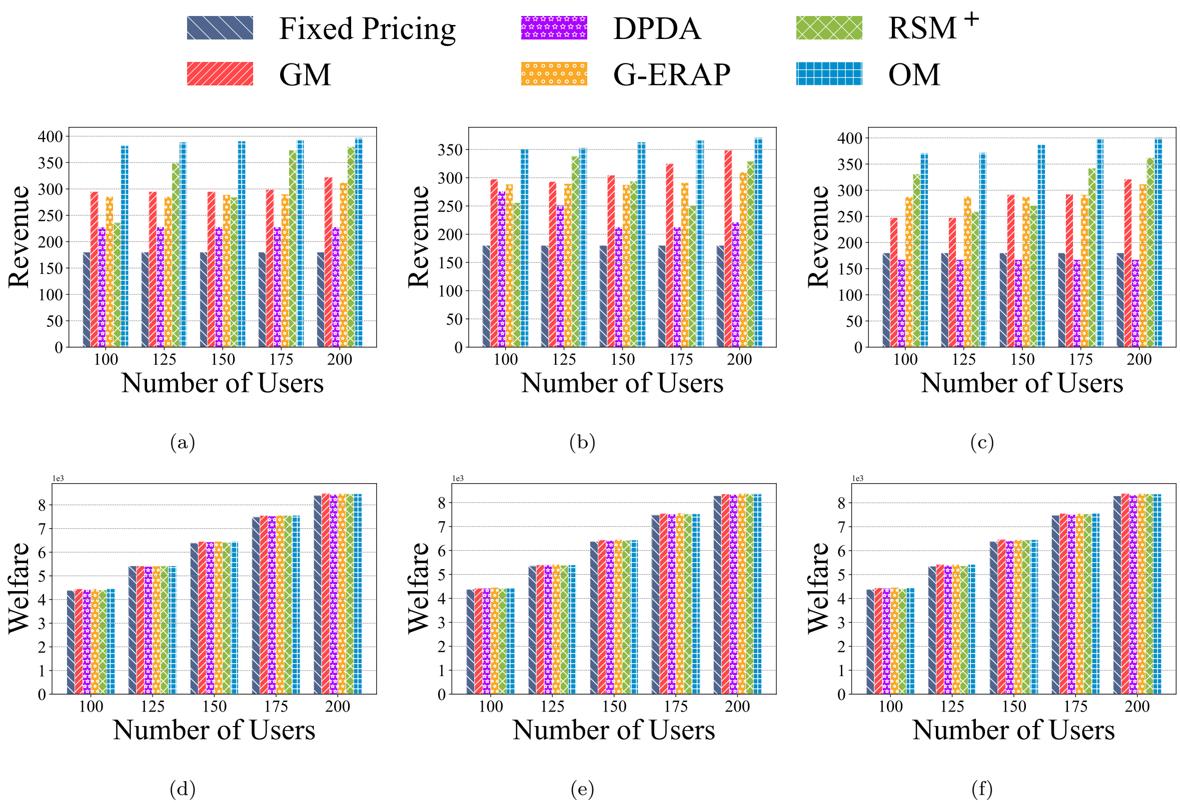


Fig.6. Impact of the distribution of the users' biases: uniform, normal and exponential distributions. (a) Revenue - uniform. (b) Revenue - normal. (c) Revenue - exponential. (d) Social welfare - uniform. (e) Social welfare - normal. (f) Social welfare - exponential.

ing and these users are excluded from serving by the edge platform. As shown in Fig. 7, when there are more users, all sampling rates can achieve higher revenue. This is straightforward since, with a certain sampling rate, more users are sampled with more users in total. And thus we can better set the price. As for different sampling rates under the same number of users, the revenue difference between different sampling rates becomes smaller when there are more users. This means that all sampling rates have sampled enough user information to set a proper price. On the other hand, when there are fewer users, both high and low sampling rates perform poorly. The reason is that when the sampling rate is high, too many users are sampled and therefore excluded, and the utilization of the resources is low; when the sampling rate is low, the sampled users' information is not enough.

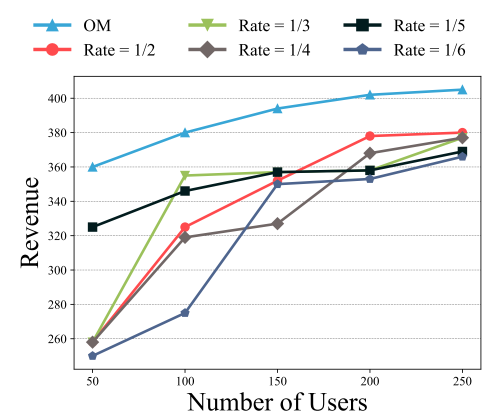


Fig.7. Impact of sampling rate.

8 Related Work

8.1 Game Theory

8.1.1 Hotelling Game

The competition between the cloud platform and the edge platform is similar to the Hotelling

Game [24] in a game theory. A general form of the Hotelling Game can be modeled as a two-stage game of complete information. There are n firms that offer products to customers. First, the firms choose product characteristics $\bar{x} = (x_1, x_2, ..., x_n) \in [0,1]^n$. Second, the firms decide the prices of the products $\bar{p} = (p_1, p_2, ..., p_n) \in \mathbb{R}^n$. The customers make decisions based on their utility functions. For customer j with preference w_j, his utility by choosing firm i is $u(w_j, p_i, x_i) = d(x_i - w_j) - p_i$, where $d(\cdot)$ is the customer's satisfaction function which depends on the distance between his preference and the product. p_i is the price set by firm i.

The Hotelling Game has received much attention. A basic extension is studying the equilibrium with different cost functions [25–31]. Thisse *et al.* [32] illustrated that the competition becomes tougher and derives a lower equilibrium price. Gabszewicz *et al.* [33] demonstrated that under price discrimination, there is a Bertrand price game at each location. Böckem amd Sabine [34] considered continuous reservation price under the uniform distribution. In this paper, platforms (firms in the hotelling model) do not choose product characteristics but only the prices for different resources. A user chooses a better platform, just like the consumer chooses the best firm.

8.1.2 Prior-free Mechanism

Without users' type information, it is impossible to design a revenue optimal mechanism. Therefore, our goal is to identify a "robust" mechanism that is competitive to a given benchmark in "all distributions" [35–38]③. Goldberg *et al.* [39] proposed the random sampling approach in designing a prior free mechanism and showed that it has a constant competitive ratio with fixed pricing in an unlimited digital good auction. Later the competitive ratio is improved in [40–42]. In this paper, we extend the random sampling approach in two directions: users have different demands of the resource instead of unit demand, and the combinatorial auctions with three different types of resources instead of a single type.

8.2 Pricing in Cloud Computing and Edge Computing

The pricing mechanism of cloud computing and edge computing is an important research topic that has been studied from various perspectives [43, 44]. Most existing studies can be classified into two categories: static pricing and dynamic pricing. Static pricing is the simplest and most common pricing mechanism, where the service provider charges a fixed price for each unit of resource. Many cloud platforms, such as AWS, Google cloud, and Microsoft Azure, adopt this mechanism. However, static pricing does not reflect the fluctuation of supply and demand, and may result in inefficient resource utilization or unfair allocation. Dynamic pricing is a more flexible and adaptive pricing mechanism, where the service provider adjusts the price according to the market condition or the user preference [45–48]. Many dynamic pricing mechanisms are based on auctions, where the users submit bids for the resources they want, and the service provider allocates the resources to the highest bidders. Zaman and Grosu [49] argued that a fixed price is not enough and propose a combinatorial auction-based allocation mechanism for resources of a single dimension. Zhang *et al.* [50] studied the problem where the cloud platform provides different types of VMs, and the users bid for bundles of VMs. The CSP performs an auction to decide the allocation based on the resources and the objective of maximizing social welfare. Shi *et al.* [51] extended the problem to an online version where the users bid bundles for some time slots. Besides auction-based mechanisms, Zhang *et al.* [52] designed a pricing mechanism based on the current resource utilization ratio.

For the edge computing scenario, Kiani and Ansari [53] proposed a hierarchical model that in-

③In this paper, the benchmark is set to be OM with all information.

troduces the concept of field, shallow, and deep cloudlets, and designed an auction-based profit maximization mechanism. Sun *et al.* [18] investigated the scenario where users have uniform resource demands and introduced DPDA, a mechanism that employs a double auction to allocate resources among multiple service providers. DPDA prioritizes users with higher bids in the allocation process. Bahreini *et al.* [19] assumed a single provider that offers VMs at both the cloud platform and the edge platform, and the users have fixed preference coefficients for different platforms. They designed G-ERAP, a greedy mechanism for resource allocation and pricing in edge systems, which sorts users by their bids divided by their resource demands, and then allocates and prices resources according to users' fixed preferences for different platforms. Jiao *et al.* [54] proposed an auction-based edge computing resource allocation mechanism, which considers allocative externalities. Li *et al.* [11] studied a three-tier edge computing market consisting of edge servers, brokers, and edge users. They proposed a pricing-based resource allocation mechanism via iterative bidding. Lyu *et al.* [55] modeled the dynamic pricing process of edge computing servers as a Markov decision process to maximize the revenue, and they propose a dynamic pricing approach based on dueling double deep Q network (D3QN). Xu *et al.* [56] used a double auction to allocate resources among different service providers, but they did not consider the case that different platforms have varying features. This paper proposes a novel pricing mechanism for edge computing that takes into account the unique features of different platforms, the competition among edge and cloud providers, and the preferences of different users. We summarize the related work for pricing in edge computing in Table 3.

Table 3. Related work for pricing in edge computing

Mechanism	Users' Resource Demand	Multi-dimensional resources	Non-Uniform Edge Bias	Edge-Cloud Competition
MECM[11]	Non-Uniform	✔	✘	✘
DPDA [18]	Uniform	✘	✔	✘
G-ERAP [19]	Non-Uniform	✔	✘	✘
Xu *et al.* [56]	Non-Uniform	✔	✘	✘
[This work]	Non-Uniform	✔	✔	✔

9 Conclusion

In this work, we proposed and studied the edge pricing game under competition (EPGC). We considered the case where the cloud platform adopts fixed prices and investigated the best mechanism for the edge platform. We proposed OM, GM, RSM, and RSM^{+} restricted to setting prices for different unit resources. GM has no theoretical guarantee on the revenue, but it works well with relatively sufficient computational resources and has the best social welfare performance. RSM has a 1/9 optimal revenue guarantee with a probability approaching 1 when no single user can contribute a large fraction to the revenue. We conducted extensive experiments on the real trace from Google cluster, and our extensive experiments validated the performance of OM, GM and RSM^{+}.

参考文献

[1] Ananthanarayanan G, Bahl P, Bodík P, Chintalapudi K, Philipose M, Ravindranath L, Sinha S. Real-time video analytics: The killer app for edge computing. *Computer*, 2017, 50(10):58–67. DOI: 10.1109/MC.2017.3641638.

[2] Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 2016, 3(5):637–646. DOI: 10.1109/JIOT.2016.2579198.

[3] Han Z, Tan H, Li X Y, Jiang S H C, Li Y, Lau F C. Ondisc: Online latency-sensitive job dispatching and scheduling in heterogeneous edge-clouds. *IEEE/ACM Trans. Networking*, 2019, 27(6):2472–2485. DOI: 10.1109/TNET.2019.2953806.

[4] Liu F, Tang G, Li Y, Cai Z, Zhang X, Zhou T. A survey on edge computing systems and tools. *Proceedings of the IEEE*, 2019, 107(8):1537–1562. DOI: 10.1109/JPROC.2019.2920341.

[5] Letaief K B, Shi Y, Lu J, Lu J. Edge artificial intelligence for 6g: Vision, enabling technologies, and applications. *IEEE Journal on Selected Areas in Communications*, 2021, 40(1):5–36. DOI: 10.1109/JSAC.2021.3126076.

[6] Al-Ansi A, Al-Ansi A M, Muthanna A, Elgendy I A, Koucheryavy A. Survey on intelligence edge computing in 6g: Characteristics, challenges, potential use cases, and market drivers. *Future Internet*, 2021, 13(5):118. DOI: 10.3390/fi13050118.

[7] Tang S, Yuan J, Li X Y. Towards optimal bidding strategy for amazon ec2 cloud spot instance. In *Proc. the 5th International Conference on Cloud Computing*, Jun. 2012, pp. 91–98. DOI: 10.1109/CLOUD.2012.134.

[8] Tang S, Yuan J, Wang C, Li X Y. A framework for amazon ec2 bidding strategy under sla constraints. *IEEE Trans. Parallel and Distributed Systems*, 2013, 25(1):2–11. DOI: 10.1109/TPDS.2013.15.

[9] Yao M, Zhang P, Li Y, Hu J, Lin C, Li X Y. Cutting your cloud computing cost for deadline-constrained batch jobs. In *Proc. the 2014 IEEE International Conference on Web Services*, Jun. 2014, pp. 337–344. DOI: 10.1109/ICWS.2014.56.

[10] Wu C, Buyya R, Ramamohanarao K. Cloud pricing models: Taxonomy, survey, and interdisciplinary challenges. *ACM Computing Surveys*, 2019, 52(6):1–36. DOI: 10.1145/3342103.

[11] Li Y, Xia M, Duan J, Chen Y. Pricing-based resource allocation in three-tier edge computing for social welfare maximization. *Computer Networks*, 2022, 217:109311. DOI: 10.1016/j.comnet.2022.109311.

[12] Tan H, Shen Z, Zihe W, Han Z, Li X. Edge-centric pricing mechanisms with selfish heterogeneous users. In *Proc. the 14th China Conference on Internet of Things (Wireless Sensor Network)*, Sept. 2020.

[13] Reiss C, Wilkes J, Hellerstein J L. Google cluster-usage traces: format + schema. Technical Report, Google Inc., Aug. 2023. https://github.com/google/cluster-data.

[14] Myerson R B. Incentive compatibility and the bargaining problem. *Econometrica: journal of the Econometric Society*, 1979, pp. 61–73. DOI: 10.2307/1912346.

[15] Li Y, Liang B, Tizghadam A. Robust online learning against malicious manipulation with application to network flow classification. In *Proc. the 2021 IEEE Conference on Computer Communications*, May 2021. DOI: 10.1109/INFOCOM42981.2021.9488890.

[16] Li Y, Liang B, Tizghadam A. Robust network flow classification against malicious feature manipulation. In *Proc. the 2020 IEEE International Conference on Communications*, Jun. 2020. DOI: 10.1109/ICC40277.2020.9148644.

[17] Li Y, Liang B, Tizghadam A. Robust online learning against malicious manipulation and feedback delay with application to network flow classification. *IEEE Journal on Selected Areas in Communications*, 2021, 39(8):2648–2663. DOI: 10.1109/jsac.2021.3087268.

[18] Sun W, Liu J, Yue Y, Zhang H. Double auction-based resource allocation for mobile edge computing in industrial internet of things. *IEEE Trans. Industrial Informatics*, 2018, 14(10):4692–4701. DOI: 10.1109/tii.2018.2855746.

[19] Bahreini T, Badri H, Grosu D. Mechanisms for resource allocation and pricing in mobile edge computing systems. *IEEE Trans. Parallel and Distributed Systems*, 2021, 33(3):667–682. DOI: 10.1109/tpds.2021.3099731.

[20] Luo Q, Hu S, Li C, Li G, Shi W. Resource scheduling in edge computing: A survey. *IEEE Communications Surveys & Tutorials*, 2021, 23(4):2131–2165. DOI: 10.1109/COMST.2021.3106401.

[21] Li B, Roy R B, Patel T, Gadepally V, Gettings K, Tiwari D. Ribbon: cost-effective and qos-aware deep learning model inference using a diverse pool of cloud computing instances. In *Proc. the 2021 International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov. 2021. DOI: 10.1145/3458817.3476168.

[22] Chen M, Hao Y. Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE Journal on Selected Areas in Communications*, 2018, 36(3):587–597. DOI: 10.1109/JSAC.2018.2815360.

[23] Du B, Wu C, Huang Z. Learning resource allocation and pricing for cloud profit maximization. In *Proc. the 2019 AAAI Conference on Artificial Intelligence*, volume 33, 2019, pp. 7570–7577. DOI: 10.1609/aaai.v33i01.33017570.

[24] Hotelling H. *The collected economics articles of Harold Hotelling*. Springer Science & Business Media, 2012.

[25] d'Aspremont C, Gabszewicz J J, Thisse J F. On hotelling's" stability in competition". *Econometrica: Journal of the Econometric Society*, 1979, pp. 1145–1150. DOI: 10.2307/1911955.

[26] Economides N. Minimal and maximal product differentiation in hotelling's duopoly. *Economics Letters*, 1986, 21(1):67–71. DOI: 10.1016/0165-1765(86)90124-2.

[27] Gabszwicz J J, Thisse J F. On the nature of competition with differentiated products. *The Economic Journal*, 1986, pp. 160–172. DOI: 10.2307/2233431.

[28] Anderson S P. Equilibrium existence in the linear model of spatial competition. *Economica*, 1988, pp. 479–491. DOI: 10.2307/2553910.

[29] Kats A. More on hotelling's stability in competition. *International Journal of Industrial Organization*, 1995, 13(1):89–93. DOI: 10.1016/0167-7187(94)00445-8.

[30] De Frutos M A, Hamoudi H, Jarque X. Equilibrium existence in the circle model with linear quadratic transport cost. *Regional Science and Urban Economics*, 1999, 29(5):605–615. DOI: 10.1016/s0166-0462(99)00014-9.

[31] Shen W, Wang Z. Hotelling-downs model with limited attraction. In *Proc. the 16th Conference on Autonomous Agents and MultiAgent Systems*, May 2017, pp. 660–668. DOI: 10.5555/3091125.3091220.

[32] Thisse J F, Vives X. Spatial pricing schemes: On the strategic choice of spatial price policy. 1998, p. 152–173. DOI: 10.1017/CBO9780511522048.009.

[33] Gabszewicz J J, Thisse J F. Location. Handbook of Game Theory with Economic Applications, pp. 281–304. Elsevier, 1992. DOI: 10.1016/S1574-0005(05)80012-8.

[34] Böckem S. A generalized model of horizontal product differentiation. *The Journal of Industrial Economics*, 1994, pp. 287–298. DOI: 10.2307/2950571.

[35] Chen N, Gravin N, Lu P. Optimal competitive auctions. In *Proc. the 46th annual ACM Symposium on Theory of Computing*, May 2014, pp. 253–262. DOI: 10.1145/2591796.2591855.

[36] Goldberg A V, Hartline J D. Competitiveness via consensus. In *Proc. the 14th annual ACM-SIAM Symposium on Discrete Algorithms*, Jan. 2003, pp. 215–222. DOI: https://dl.acm.org/doi/10.5555/644108.644145.

[37] Devanur N R, Hartline J D. Limited and online supply and the bayesian foundations of prior-free mechanism design. In *Proc. the 10th ACM Conference on Electronic Commerce*, Jul. 2009, pp. 41–50. DOI: 10.1145/1566374.1566381.

[38] Ichiba T, Iwama K. Averaging techniques for competitive auctions. In *Proc. the 7th Workshop on Analytic Algorithmics and Combinatorics*, Jan. 2010, pp. 74–81. DOI: 10.1137/1.9781611973006.10.

[39] Goldberg A V, Hartline J D, Wright A. Competitive auctions and digital goods. In *Proc. the 12th annual ACM-SIAM symposium on Discrete algorithms*, Jan. 2001, pp. 735–744. DOI: https://dl.acm.org/doi/10.5555/365411.365768.

[40] Alaei S, Malekian A, Srinivasan A. On random sampling auctions for digital goods. *ACM Trans. Economics and Computation*, 2014, 2(3):1–19. DOI: 10.1145/2517148.

[41] Feige U, Flaxman A, Hartline J D, Kleinberg R. On the competitive ratio of the random sampling auction. In *Lecture Notes in Computer Science 3828*, 2005, pp. 878–886. DOI: 10.1007/11600930_89.

[42] Fiat A, Goldberg A V, Hartline J D, Karlin A R. Competitive generalized auctions. In *Proc. the 34th annual ACM Symposium on Theory of Computing*, May 2002, pp. 72–81. DOI: 10.1145/509907.509921.

[43] Wang Q, Ren K, Meng X. When cloud meets ebay: Towards effective pricing for cloud computing. In *Proc. the 2012 IEEE International Conference on Computer Communications*, May 2012, pp. 936–944. DOI: 10.1109/INFCOM.2012.6195844.

[44] Wang Q, Guo S, Liu J, Pan C, Yang L. Profit maximization incentive mechanism for resource providers in mobile edge computing. *IEEE Trans. Services Computing*, 2019, 15(1):138–149. DOI: 10.1109/TSC.2019.2924002.

[45] Lehmann S, Buxmann P. Pricing strategies of software vendors. *Business & Information Systems Engineering*, 2009, 1(6):452. DOI: 10.1007/s12599-009-0075-y.

[46] Ruiz-Agundez I, Penya Y K, Bringas P G. A flexible accounting model for cloud computing. In *Proc. the 2011 Annual SRII Global Conference*, Mar. 2011, pp. 277–284. DOI: 10.1109/SRII.2011.38.

[47] Macías M, Guitart J. A genetic model for pricing in cloud computing markets. In *Proc. the 2011 ACM Symposium on Applied Computing*, Mar. 2011, pp. 113–118. DOI: 1982185.1982216.

[48] Sharma B, Thulasiram R K, Thulasiraman P, Garg S K, Buyya R. Pricing cloud compute commodities: A novel financial economic model. In *Proc. the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, May 2012, pp. 451–457. DOI: 10.1109/CCGrid.2012.126.

[49] Zaman S, Grosu D. Combinatorial auction-based allocation of virtual machine instances

in clouds. *Journal of Parallel and Distributed Computing*, 2013, 73(4):495–508. DOI: 10.1016/j.jpdc.2012.12.006.

[50] Zhang L, Li Z, Wu C. Dynamic resource provisioning in cloud computing: A randomized auction approach. In *Proc. the 2014 IEEE International Conference on Computer Communications*, Apr. 2014, pp. 433–441. DOI: 10.1109/INFOCOM.2014.6847966.

[51] Shi W, Zhang L, Wu C, Li Z, Lau F. An online auction framework for dynamic resource provisioning in cloud computing. In *ACM SIGMETRICS Performance Evaluation Review*, volume 42, 2014, pp. 71–83. DOI: 10.1145/2591971.2591980.

[52] Zhang Z, Li Z, Wu C. Optimal posted prices for online cloud resource allocation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2017, 1(1):23. DOI: 10.1145/3084460.

[53] Kiani A, Ansari N. Toward hierarchical mobile edge computing: An auction-based profit maximization approach. *IEEE Internet of Things Journal*, 2017, 4(6):2082–2091. DOI: 10.1109/JIOT.2017.2750030.

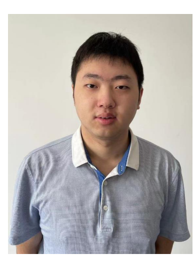
[54] Jiao Y, Wang P, Niyato D, Suankaewmanee K. Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks. *IEEE Trans. Parallel and Distributed Systems*, 2019, 30(9):1975–1989. DOI: 10.1109/TPDS.2019.2900238.

[55] Lyu F, Cai X, Wu F, Lu H, Duan S, Ren J. Dynamic pricing scheme for edge computing services: A two-layer reinforcement learning approach. In *Proc. the 30th IEEE/ACM International Symposium on Quality of Service*, Mar. 2022. DOI: 10.1109/IWQoS54832.2022.9812869.

[56] Xu K, Zhang Y, Shi X, Wang H, Wang Y, Shen M. Online combinatorial double auction for mobile cloud computing markets. In *Proc. the 33rd IEEE International Performance Computing and Communications Conference*, Dec. 2014. DOI: 10.1109/PCCC.2014.7017103.

Hai-Sheng Tan is currently a professor at University of Science and Technology of China (USTC), Hefei. He received his B.E. degree in software engineering and B.S. degree in management both from USTC with the highest honor. Then, he got his Ph.D. degree in computer science at the University of Hong Kong (HKU), Hong Kong. His research interests lie primarily in networking algorithm design and system implementation, where he has published over 90 papers in prestigious journals and conferences. He recently received the Best Paper Award in WASA'19, CWSN'20, PDCAT'20 and ICPADS' 21. He is a senior member of IEEE.

Guo-Peng Li is a Ph.D. candidate at University of Science and Technology of China (USTC), Hefei. He received his B.E. degree in computer science and technology from Central South University, Changsha, in 2020. His research interests are edge computing and resource-efficient AI.



Zi-Yu Shen received his B.E. and M.E. degrees in computer science and technology from University of Science and Technology of China (USTC), Hefei. His research interests are edge computing and mechanism design.

Zi-He Wang is currently an assistant professor with the Gaoling School of artificial intelligence, Renmin University of China, Beijing. He received his B.E. and Ph.D. degrees in computer science from Tsinghua University, Beijing, in 2011 and 2016, respectively. He was an assistant professor with the Shanghai University of Finance and Economics. His research focuses on algorithms and mechanism design.

Zhen-Hua Han is a senior researcher at Microsoft Research (Asia), Shanghai. He received B.E. degree in electronic and information engineering from University of Electronic Science and Technology of China, Chengdu, in 2014, and Ph.D. degree from the University of Hong Kong (HKU), Hong Kong. His research interests are resource management, systems for machine learning, and cloud computing. Many of his works have been published in top venues such as USENIX OSDI, ACM SOSP, ASPLOS, and IEEE INFOCOM.

Ming-Jun Xiao is a professor at University of Science and Technology of China (USTC), Hefei. He received his Ph.D. degree from USTC in 2004. His research interests include mobile crowdsensing, edge computing, federated learning, auction theory, data security and privacy. He has published more over 100 papers in referred journals and conferences, including TMC, TC, TPDS, TON, TKDE, TSC, INFOCOM, ICDE, ICDCS, ICNP, etc. He served as the TPC member of INFOCOM, IJCAI, ICDCS, DASFAA and etc. He is on the reviewer board of several top journals such as TMC, TON, TPDS, TSC, TVT, TCC, etc.

Xiang-Yang Li is a full professor and executive dean at school of computer science and technology, USTC, Hefei, China. He was a full professor at Illinois Institute of Technology, Chicago, USA. He is an ACM/IEEE Fellow and an ACM distinguished scientist. Dr. Li received M.S. (2000) and Ph.D. (2001) degree at the department of computer science from University of Illinois at Urbana-Champaign, a Bachelor degree at department of computer science and a Bachelor degree at department of business management from Tsinghua University, both in 1995. His research spans artificial intelligent Internet of things, mobile computing, data sharing and trading, and privacy. He published a monograph "Wireless Ad Hoc and Sensor Networks: Theory and Applications".

Guo-Liang Chen received the B.S. degree from Xi'an Jiaotong University, Xi'an, in 1961. Since 1973, he has been with the University of Science and Technology of China, Hefei, a professor in computer science. From 1981 to 1983, he was a visiting scholar at Purdue University, West Lafayette, IN. He has published 9 books and more than 200 research papers. His research interests include parallel algorithms, computer architectures, computer networks, and computational intelligence. He is an academician of the Chinese Academy of Sciences.