Edge-Centric Pricing Mechanisms with Selfish Heterogeneous Users

Haisheng Tan[1], Ziyu Shen[1], Zihe Wang[2], Zhenhua Han[3], and Xiangyang Li[1]

[1] *USTC LINKE Lab*
hstan@ustc.edu.cn, shenziyu@mail.ustc.edu.cn xiangyangli@ustc.edu.cn
[2] *Shanghai University of Finance and Economics*
wang.zihe@mail.shufe.edu.cn
[3] *Microsoft Research Asia*
Zhenhua.Han@microsoft.com

Abstract

Through deploying computing resource 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 in a non-cooperative game, which set prices for computational resources to attract users with different real-time requirements. In this work, we investigate the truthful pricing mechanisms at the edge competing with the cloud under three different settings. When full user information is available, the optimal mechanism can be achieved based on a knapsack problem oracle. With partial information, where users' resource demand is given but his preference information to the edge is private, we propose a random sampling mechanism that achieves a constant approximation with probability approaching one. We also propose an efficient heuristic greedy mechanism. Both mechanisms are truthful, which can be extended to the prior-free setting where all users' information is private. Finally, extensive simulations are conducted on the Google cluster dataset. The results validate our theoretical analysis that the greedy mechanism works well in the market where edge resource is scarce, while the random sampling mechanism performs better when the edge platform has a larger capacity constraint.

1 Introduction

Internet of Things (IoT) and mobile computing have been developing rapidly and drawn extensive attention from both academia and industry. Applications like voice assistant need to perform computation-intensive speech recognition while providing services to users. To overcome the limitations of computation resources and energy, applications often offload these tasks to the remote data center. This computing paradigm is known as Mobile Cloud Computing (MCC), which has enabled a lot of convenient services. However, some emerging applications like augmented reality or autonomous driving require real-time video processing [1]. MCC's long transmission latency brought by geographical distance between the data source 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 in close proximity to the users. Edge computing can respond to real-time applications in a timely manner and is regarded as the key technology to achieve the 5G vision [2].

The development of edge computing and the deployment of edge servers need to be promoted with economic benefits, so it is vital to develop a suitable pricing mechanism. There have been many pieces of research that focus on the pricing mechanism of cloud computing. Static pricing is the most widely adopted pricing mechanism [3–5], and dynamic pricing is studied in many different perspectives [6–9]. These pricing mechanisms imply a hypothesis that the cloud has unlimited computing resources. More recent researches focus on both pricing mechanism and resource allocation, and many auction based mechanisms are proposed. Zaman and Grosu [10] argue that fixed price is not enough and proposed a combinatorial auction based allocation mechanism for resources of a single dimension. Zhang et al. [11] study the problem under the setting that the cloud platform provides some different types of VMs, and the users bid for bundles of VMs. The cloud service provider performs an auction to decide the allocation based on the resources and the objective of maximizing social welfare. And then they further extend the problem to an online version that the user bids bundles for some time slots [12]. Zhang et al. [13] study a more flexible model that users can choose the amount of resources and assemble VMs dynamically. An auction determines the allocation, and the users will pay the cloud platform based on the resource they occupied. Zhang et al. [14] consider three different user utility types and transform different types into different bids and then perform an auction. Besides auction based mechanisms, Zhang et al. [15] design a pricing mechanism based on the current resource utilization ratio.

However, seldom existing research focused on the edge computing scenario, while the pricing mechanism of the edge platform is quite different from the mechanism of the cloud platform. First, compared with the cloud platform, edge servers have relatively limited computational resources, so we have to carefully design the pricing mechanism to maximize the revenue. Besides, even when the optimal prices are given, the optimal allocation rule that determines which tasks should be executed on the edge platform is a hard problem. Second, there is a competition between the edge and the cloud platform. Therefore, the pricing mechanism of the edge platform is constrained by the cloud. A proper charging mechanism should be adopted at the edge, or else the user would choose the cloud platform to execute the task. Finally, different users have different sensitivity to the response time. Therefore they have different preferences to choose the cloud or the edge platform. It is crucial to use this property to maximize revenue for edge computing. The above characteristics make it challenging to design the pricing mechanism for the edge. Kiani and Ansari [16] proposed a hierarchical model by introducing the concept of field, shallow, deep cloudlets, and designed an auction-based profit maximization mechanism. Bahreini et al. [17] assume a single provider who provides VMs at both the cloud and the edge platform. The users have a fixed preference coefficient of the utility of different platforms, and the allocation is determined by an envy-free auction mechanism. Most researches above assume a single service provider; however, there are usually multiple service providers. Xu et al. [18] and Sun et al. [19] use a double auction to allocate the resources, but they did not consider the case that different platforms have different strengths.

In this paper, we propose the Edge Pricing Game under Competition (EPGC), and study the pricing mechanism of the edge platform. Our objective is to maximize the income of the edge platform. Each user has some computation resource requirement. The edge platform will decide the pricing of resources, and the users will choose to purchase resources at the edge or the cloud. To fit the actual scenario, we assume the following general settings: 1) Multi-dimension 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 requests [20]; 3) Indivisible computational resources: a user's required computational resources should all be allocated at the cloud or the edge platform, but not both. If a user's requirement can be allocated at both platforms, we will divide the case into two different users; and 4) Bias of the edge platform: to model the latency requirement of different applications, we set each user has a personal preference level to the edge platform which is of lower latency. If the total charge of the edge platform is higher than the cloud by no more than the preference value, the user will tend to choose the edge platform.

To our best knowledge, this is the first work to study the pricing mechanism for edge computing taking into account the competition of the cloud. Our contributions can be summarized as follows:

- We investigate the optimal pricing mechanism for the edge platform against the cloud. Three settings are considered depending on how much information the platforms know about the users, i.e., full, partial, and none of the user information. We focus on designing truthful mechanisms with no discriminate prices. Specifically, with full user information, the optimal mechanism is achieved (Sec. 3). The random sampling mechanism is competitive that guarantees 1/9 optimal revenue with probability approaching 1 with partial user information (Sec. 4). Both the above mechanisms can be extended when no user information is available(Sec. 5)

- We evaluate different mechanisms by extensive simulations based on the data-trace from Google Cluster [21]. Simulation results show that our mechanisms perform consistently well to maximize the revenue with different settings of various parameters. Most notably, the random sampling mechanism can achieve near-optimal revenue of the edge platform. (Sec. 6).

2 System and Problem Definition

Network Model: We consider single time slot network. 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 by 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. As for the amount of available computational resources, the edge platform has relatively limited computational resources, and the cloud platform has unlimited amount of resources. The available amount of resources of different types at the edge platform is denoted as $C_e^g, g = 1, 2, 3$.

The utility 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 cost is not zero.

Pricing Model: The relationship between the edge and the cloud platforms is competitive. Both seek to maximize their own revenue by renting their computational resources to users to execute their tasks. In practice, the scale of the cloud platform is typically much larger than the edge. We assume the cloud platform sets prices p_c^g for using a unit of computational resources of type g at first. The edge platform then decides the prices of three different resources p_e^g to maximize edge platform's revenue in response to cloud prices.

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. The revenue of the cloud and the edge platforms are denoted as u_c and u_e, respectively.

User Model: There are n users, and each user has a computation task to execute at the cloud or the edge platform. The resources required by user i is denoted as d_i^g, the edge or the cloud platform will pack the required resources as a VM to execute the request of the user. Given two price profiles $\{p_e^g\}$ and $\{p_c^g\}$, each user will choose to purchase resources from one platform. In the edge computing scenario, the edge platform can complete computation tasks with lower latency than the cloud platform. So the users prefer the edge platform to execute their tasks. We model this preference by incurring bias v_i of user i. That is to say, if the edge platform charges the user i more than the cloud platform charges the user i by no more than v_i, the user will choose the edge platform. Specifically, user i is characterized by vector $(d_i^1, ..., d_i^\tau, v_i)$, and he will choose the edge platform if

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

Even user i is willing to choose edge platform given the prices, it is possible that user i is not chosen by the edge platform due to the capacity constraint. The revenue of edge platform can be denoted as: $u_e = \sum_{i \in E} \sum_{g=1}^{\tau} p_e^g d_i^g$, where E is the set of users whose computation tasks are executed at the edge platform.

Problem Formulation: Based on the settings above, we formulate the Edge Pricing Game under Competition (EPGC).

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

We focus on the pricing strategies. When designing mechanisms, we take both the pricing and choosing candidate strategies into consideration.

3 Full User Information

In this section, we consider the mechanism design with full information. Every user's type is public information which includes both the demand and the bias between two platforms. In a mechanism, we need to specify the set of users chosen by the edge and what the prices are. We focus on designing a "fair" mechanism that we do not set discriminate prices for different users, and the payment is proportional to how much resources a user rents on the edge platform. We show that designing the optimal mechanism is hard.

Theorem 1. *Designing the optimal mechanism for the edge, even with a fixed cloud pricing strategy, is NP-hard.*

Proof. The proof is based on 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 edge platform is $(s, 0, 0)$. Obviously, the optimal prices are $(1, 0, 0)$. The values per unit resource of all users 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.

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 indifferent from two platforms.*

The proof is omitted, and the idea is as follows. If there are less than $\tau - k$ users who are indifferent from two platforms, we can always adjust the prices and increase the edge's revenue.

Mechanism 1 describes the optimal mechanism of the edge platform. It first enumerates a set of users who are indifferent choosing 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 edge and cloud price plans such that the equities of in eqn. (1) holds for the chosen users, and there are

Mechanism 1: Optimal Mechanism with Full Information

1 **Input** $(d_i^g, v_i), g = 1, ..., \tau, i = 1, ..., n; C_e^g, g = 1, ..., \tau$;
2 $u_e = 0$,$A = \emptyset$
3 **Function** `OptimalAssignment(`S`)`:
4   **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**
5         $(A, u_e') =$`OptimalKnapsack`$(\{(\{d_i^g\}, v_i), i \in S\}, \{C_e^g\})$
6   **if** $u_e' > u_e$ **then**
7             $u_e = u_e'$
8             $\{p_e'^g\} = \{p_e^g\}$
9   **return** $A, u_e, \{p_e'^g\}$
10 $u_{opt} = 0$
11 **for** *all combinations* S *of three users* i, j, k*; two users* i, j *and one user* i **do**
12     $(A, u_e, \{p_e'^g\}) =$ `OptimalAssignment(`S`)`
13   **if** $u_{opt} < u_e$ **then**
14         $u_{opt} = u_e$
15         $\{p_e^{opt,g}\} = \{p_e'^g\}$
16         $A_{opt} = A$
17 **return** $A_{opt}, \{p_e^{opt,g}\}$

zero/one/two types of resources' whose price is 0 (Line 4). Then, with each fixed price plan, we can use three-dimensional knapsack oracle to find the optimal set of users to be placed on the edge platform (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.

4 Partial User Information

In this section, we consider the case that a user's demand is public, while his bias information is private. We would like the mechanism to be competitive, i.e., 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 little fraction to the optimal revenue.

We consider the ex-post truthful mechanism. We first solicit users' bids, and users will truthfully report their bias. Then we set prices for three resources and determine who would be chosen by the edge platform. There are two basic ideas to set prices: one is to set the equilibrium prices when supply equals the demand, and the other is to set more reasonable prices by learning the distribution of the bias.

4.1 Greedy Mechanism

Following the first idea, we propose a greedy mechanism. Such mechanisms perform well in the scarce market where the supply is far smaller than the total demand of users. We first normalize capacities for the three resources to be the same. Greedy mechanism 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 $\frac{v_i}{d_i^1+d_i^2+d_i^3}$.

Mechanism 2: Greedy Mechanism

Input $(d_i^g), g = 1, ..., \tau, i = 1, ..., n; C_e^g, g = 1, ..., \tau$;
All users report their bias $v_i, i = 1, ..., n$
$S = \emptyset$
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,$\{p_c^g + q_{m(k)}\}$

The greedy mechanism takes the demand as input and collect users' private information. At last, it outputs the set of users chosen by the edge platform and corresponding prices for different type of resources.

This greedy mechanism works well if the demands for three resources are not correlated heavily. So the consumption of the resources is balanced in the outcome. In the case that most of the high-valued users' demand focus on the same type of resource, this greedy mechanism performs terrible. Consider the following example.

Example 1. *There are 7 users. The first three only need the first type of resources. The other four users need all three types. Particularly, for users* $i = 1, 2, 3$*,* $(d_i^1, d_i^2, d_i^3, v_i) = (1, 0, 0, 1)$*, 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 user $1, 2, 3$ and set prices $(1/3, 1/3, 1/3)$ applying our greedy mechanism, and the revenue is 3. Since we have three types of resources and the user 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 user $4, 5, 6$ and set prices $(2.9/3, 2.9/3, 2.9/3)$ applying the modified greedy mechanism, and the revenue would be 8.7.

Theorem 2. *The mechanism setting the same prices for three resources can only guarantee $1/n$ fraction of the optimal revenue, where n is the number of users.*

The proof is in Appendix A. Since greedy mechanism sets the same prices for three resources, so we claim that

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

Our greedy mechanism has a good welfare guarantee. Welfare is defined as the sum of biases of users chosen by edge platform. In other words, welfare is the social improvement due to the existence of the edge platform.

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

Proof. We use the same notations as in the greedy mechanism. Assume the edge platform chooses the first k users: $m(1), ..., m(k)$. The total welfare is then $\sum_{i=1,...,k} v_{m(i)}$. When the largest welfare is achieved, we denote the set of users on the edge platform by $h(1), ..., h(l)$.

By definition, user $m(k+1)$ does not fit in the edge platform due to the limited capacity. We design a virtual user with a task similar as user $m(k+1)$ but with a smaller scale such that exactly fit in the remaining capacity of the edge platform. Formally, virtual user $n+1$ has type $(\theta^* d^1_{m(k+1)}, \theta^* d^2_{m(k+1)}, \theta^* d^3_{m(k+1)}, \theta^* v^1_{m(k+1)})$ where $\theta^* = \max\{\theta | \theta d^1_{m(k+1)} + \sum_{i=1,...,k} d_i^g \leq C_e, g = 1, 2, 3\}$. Then we have

$$\begin{aligned} & \frac{v_{n+1} + \sum_{i=1,...,k} v_{m(i)}}{C_e} \\ \geq & \frac{\sum_{i=1,...,k+1} v_{m(i)}}{\sum_{i=1,...,k+1}(d^1_{m(i)} + d^2_{m(i)} + d^3_{m(i)})} \\ \geq & \frac{\sum_{i=1,...,l} v_{h(i)}}{\sum_{i=1,...,l}(d^1_{h(i)} + d^2_{h(i)} + d^3_{h(i)})} \\ \geq & \frac{\sum_{i=1,...,l} v_{h(i)}}{3C_e} \end{aligned}$$

By assumption, we have $v_{m(k+1)} \leq \frac{1}{\beta} \sum_{i=1,...,l} v_{h(i)} \leq \frac{3}{\beta} \sum_{i=1,...,k+1} v_{m(i)}$. Hence the total welfare achieved by greedy mechanism is

$$\begin{aligned} \textstyle\sum_{i=1,...,k+1} v_{m(i)} & \geq (1 - \frac{3}{\beta}) \sum_{i=1,...,k+1} v_{m(i)} \\ & \geq (\frac{1}{3} - \frac{1}{\beta}) \sum_{i=1,...,l} v_{h(i)} \end{aligned}$$

□

4.2 Random Sampling Mechanism

In the mechanism, after all users report their valuations, we first sample some users out. Then we learn their valuations and compute the optimal prices on the samples. Third, we apply the optimal prices to the remaining users. At last, we solve a knapsack problem.

Mechanism 3: Random Sampling

Input $(d_i^g), g = 1, ..., \tau, i = 1, ..., n; C_e^g, g = 1, ..., \tau;$
All users report their bias $v_i, i = 1, ..., n$
$S = \emptyset$
for *all users i* **do**
  With probability $\frac{1}{2}$, $S = S \cup \{i\}$
$(S_1, \{p_e^g\}) = Mechanism1(\{(\{d_i^g\}, v_i), i \in S\}, \{C_e^g\})$
$T = \{i \in N \backslash S | v_i \geq \sum_{g=1,...,r} p_g * d_i^g\}$
$(A, u_e) =$`Knapsack`$(\{(\{d_i^g\}, v_i), i \in T\}, \{C_e^g\})$
return $A, \{p_e^g\}$

Theorem 4. *The random sampling mechanism is ex-post truthful.*

The proof is in Appendix B. Next, we show a lemma that would be used in the proof of random sampling mechanism is competitive.

Lemma 2. *For any i, x_i is a random varible 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 C. We run a random sampling mechanism. 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 full 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 5. *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 sampling mechanism.*

Proof. A_N is the set of users chosen for N with full information. Thus we have

$$\max_{i\in A_N}\{\sum_{g=1,2,3} d_i^g p_N^g\} \le \frac{1}{\beta}R_N \le \frac{1}{\beta}\sum_{i\in A}\sum_{g=1,2,3} d_i^g p_N^g.$$

By sampling method, we divide A_N into approximately equal size with high probability. Particularly, we define random varibles for users in A_N. For user $i \in A_N$, if $i \in S$ we set $x_i = \sum_{g=1,2,3} d_i^g p_N^g$ otherwise $x_i = 0$.

By Lemma 2, we have $\Pr[\sum_{i\in A_N} x_i \notin [\frac{R_N}{3}, \frac{2R_N}{3}]] \le 2\exp(-\frac{\beta}{18})$. So with probability at least $1 - 2\exp(-\frac{\beta}{18})$, we have $\sum_{i\in A_N} x_i \in [\frac{R_N}{3}, \frac{2R_N}{3}]$. By using price $p_N^g, g = 1,2,3$ on samples we can guarantee $\frac{R_N}{3}$ revenue, so $R_S \ge \frac{R_N}{3}$.

We call a set of users T is good with prices $p^g, g = 1,2,3$ if the following two conditions hold:

- $\sum_{i\in T}\sum_{g=1,2,3} d_i^g p^g \ge \frac{R_N}{3}$
- $\sum_{g=1,2,3} d_i^g \le C_e^g$, i.e., users in T fit in the edge platform.

Similarly, we define random varibles for users in T. For user $i \in T$, if $i \in S$ we set $y_i = \sum_{g=1,2,3} d_i^g p^g$ otherwise $x_i = 0$. $t_i \le \frac{1}{\beta}R_N \le \frac{3}{\beta}\sum_{i\in T}\sum_{g=1,2,3} d_i^g p^g$.

By applying Lemma 2, we have

$$\begin{aligned}
&\Pr[\sum_{i\in T} y_i \le \frac{R_N}{9}] \\
\le\ &\Pr[\sum_{i\in T} y_i \notin [\frac{\sum_{i\in T}\sum_g d_i^g p^g}{3}, \frac{2\sum_{i\in T}\sum_g d_i^g p^g}{3}]] \\
\le\ &2\exp(-\frac{\beta}{54})
\end{aligned}$$

Thus by using prices $p^g, g = 1,2,3$, $T\backslash S$ has revenue at least $\frac{R_N}{9}$ with at least $1 - 2\exp(-\frac{\beta}{54})$.

When we compute the optimal prices on sampling users with full information, the number of possible prices is tiny compared with the number of different samplings. When introducing Mechanism 1, we have shown the optimal prices have only $q = \binom{n}{3} + 3\binom{n}{2} + 3\binom{n}{1} = \frac{n^3+6n^2+11n}{6} \le \frac{n^3}{2}$ possibilities, denoted by $(p_i^g, g = 1,2,3), i = 1,...,q$.

Then we define the set W_i as the set of users who can afford prices $p_i^g, g = 1,2,3$. If W_i has a good subset, then we call W_i wonderful set and let T_i denote the corresponding good subset.

We have shown that with probability at least $1 - 2\exp(-\frac{\beta}{18})$, we will choose prices $(p_i^g, g = 1,2,3)$ such that W_i is wonderful. We have also shown that with probability at least $1 - 2\exp(-\frac{\beta}{54})$, $T_i\backslash S$ gives revenue at least $\frac{R_N}{9}$. By union bound, with probability at least $1 - 2q\exp(-\frac{\beta}{54}) - 2\exp(-\frac{\beta}{18})$, all $T_i\backslash S, i = 1,...,q$ give revenue at least $\frac{R_N}{9}$ and the mechanism chooses the prices of some wonderful set. In other words, a random sampling mechanism guarantees $\frac{R_N}{9}$ revenue with probability approaching 1. ☐

5 No User Information

In this section, we consider a user's demand information and bias information are both private. If a user misreports demand information, he will only be interested in revealing greater demand. Otherwise, he is not able to accomplish the task even he has been chosen by the edge platform. We still focus on the truthful mechanism and consider if the mechanism introduced in the previous section can be used in this setting.

Theorem 6. *The greedy mechanism is truthful.*

Proof. To prove the mechanism is truthful, it is sufficient to prove a user's dominant strategy is revealing his type truthfully no matter what other users report. Suppose user i's true type is $(d_i^1, d_i^2, d_i^3, v_i)$ then his per unit price is $\frac{v_i}{d_i^1+d_i^2+d_i^3}$. We denote user i misreports his type as $(d_i^{1\prime}, d_i^{2\prime}, d_i^{3\prime}, v_i)$. We denote user i's ranking for truthful report and misreport as k_1 and k_2 respectively. There are two possibilities how user i's ranking changes.

- User i's ranking becomes lower, i.e., $k_2 > k_1$. It implies user i's per-unit price decreases. When we run greedy algorithms, if the execution stops before reaching user i, then user i's utility becomes zero. If the execution stops after reaching user i and does not choose user i, then user i's utility is still zero. If the algorithm chooses user i with ranking k_2, then the algorithm must choose user i with ranking k_1. We notice that the greedy algorithm must stop after checking the user with ranking k_2 when user i has ranking k_1. Furthermore, since the user will only report greater demand, the algorithm will stop at the same or earlier position. More importantly, the prices will be higher if he misreports. Thus user i's payment will stay the same or increase.
- User i's ranking becomes higher, i.e., $k_2 \le k_1$. There are two subcases.
  - User i with ranking k_1 is chosen by the mechanism. Similarly to the argument in the first possibility, the unit price never decreases. Since the fake demand can only be larger, user i's payment never decreases.
  - User i with ranking k_1 is not chosen by the mechanism. When i misreports, the greedy algorithm will stop before reaching the user with ranking k_1. So the per-unit price set by the greedy algorithm will be higher than $\frac{v_i}{d_i^1+d_i^2+d_i^3}$. That means user i's utility would be non-positive.

In summary, user i cannot increase his utility by misreporting. The mechanism is truthful. □

We also observe that the random sampling mechanism 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 our random sampling mechanism, user 2 will be chosen by edge platform since user 2 has 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 the random sampling mechanism fails to be truthful is that users are chosen using a knapsack algorithm, which depends on the user's demand. We propose a modified mechanism, where we first arrange the users in uniformly random order and then accept user sequentially if he fits on edge platform.

Mechanism 4: Modified Random Sampling

Input
  $(d_i^g), g = 1, ..., \tau, i = 1, ..., n; C_e^g, g = 1, ..., \tau;$
All users report their types $\{(d_i^g, v_i)\}$
$S = \emptyset$
for *all users* i **do**
  $S = S \cup \{i\}$ with probability $\frac{1}{2}$
$(S_1, \{p_e^g\}) = Mechanism1(\{(\{d_i^g\}, v_i), i \in S\}, \{C_e^g\})$
$T = \{i \in N \backslash S | v_i \geq \sum_{g=1,...,\tau} p_e^g * d_i^g\}$
$A = \emptyset$
for $i \in T$ *in random order* **do**
  if $d_i^g + \sum_{j \in A} d_j^g \leq C_e^g$ *for* $g = 1, 2, 3$ **then**
    $A = A \cup \{i\}$
return $A, \{p_e^g\}$

Theorem 7. *The modified mechanism 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. So the mechanism is truthful. □

6 Evaluation

In this section, we evaluate the proposed greedy and random sampling mechanisms by extensive simulations on a real-world data-trace.

6.1 Experiment Settings

We use the data set of Google cluster [21] to obtain the resource requirement of tasks. We group the tasks by their release time and divide them into different time slots. We choose the optimal pricing mechanism of the edge platform as a baseline, and need to solve 3-dimension 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, we use the pricing of Google Cloud [20]. For ease of representation, we normalize the price to 280 per CPU core, 35 per GB of memory, 1 per GB of disk, and the capacity of the edge platform of all resources are normalized to 1 by default. There is no existing data of the bias. Therefore, the bias of each task is generated randomly from 1 to 10 by default.

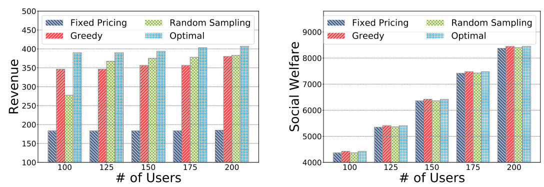
6.2 Mechanism and Metrics

Under the experiment settings above, we implement and compare the optimal mechanism (Mechanism 1, represented by "Optimal"), greedy mechanism (Mechanism 2, represented by "Greedy"), and the random sampling mechanism (Mechanism 4, represented by "Random Sampling"). 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 the random sampling mechanism, 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.

6.3 Simulation Results

Here, we present the simulation results on the performance of the pricing mechanisms on impact of the number of users, the capacity of the edge platform, the bias of users, and the sampling rate.

6.3.1 The Impact of Number of Users



(a) Revenue (b) Social Welfare

Figure 1: Impact of Number of Users

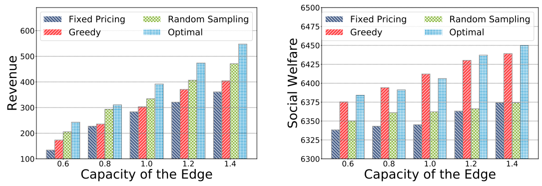
Fig. 1 demonstrates the revenue of the edge platform and the social welfare with the number of users from 100 to 200. As shown in the figure, with the increment of the number of users, the optimal revenue of the edge platform doesn't increase.

This is due to limited computational resources. When there are more users, the random sampling mechanism samples more users for pricing and therefore can set better prices to maximize the revenue. Unlike the random sampling mechanism, the revenue of the greedy mechanism performs arbitrarily with the number of users increases.

It should be noted that when there are 100 users, the greedy mechanism performs better than the random sampling mechanism. The reason is that the random sampling mechanism excludes half of all the users (50 in this case), this will cause insufficient utilization of the computational resources at the edge platform. This shortcoming can be overcome by adjusting the sampling rate, which will be discussed in the parts behind.

As for the social welfare, it is mainly influenced by the number of users. The greedy mechanism achieves the best. The reason is that the greedy mechanism selects users with the higher bias.

6.3.2 The Impact of the Capacity of the Edge Platform



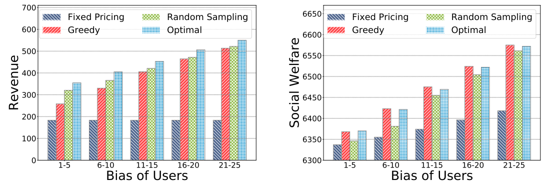
(a) Revenue (b) Social Welfare

Figure 2: Impact of Capacity of the Edge Platform

Since we have performed normalization to 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 increases or decreases. Fig. 2 demonstrates the revenue of the edge with the capacity of three resources from 0.6 to 1.4. As we can see, the revenue of all the three mechanisms increases as the capacity of the edge platform increases. When the capacity reaches 1.4, similar to the situation when there are 100 users, the random sampling performs worse than the greedy due to insufficient resource utilization. As for the social welfare, the result is similar. However, the increment of capacity of the edge doesn't result in significant increase in social welfare. The reason is that the cloud platform can contribute to the social welfare when a user's request isn't served by the edge.

6.3.3 The Impact of Bias of Users

Fig. 3 demonstrates the revenue of the edge platform of different ranges of bias of users. We simulate 5 group of users, having bias of 1 to 5, 6 to 10, 11 to 15, 16 to 20 and 21 to 25 respectively. The



(a) Revenue (b) Social Welfare

Figure 3: Impact of Bias of Users

simulation result is quite straightforward that with higher bias, the users are willing to pay more for executing tasks at the edge. This leads to higher pricing for a unit of computational resource, and therefore the revenue becomes higher.

6.3.4 The Impact of Sampling Rate

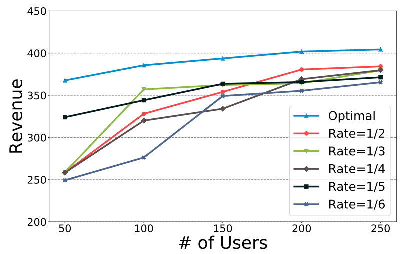


Figure 4: Impact of Sampling Rate

We investigate the revenue of the edge platform with the random sampling mechanism under different sampling rates. As shown in Fig. 4, when there are more users, all sampling rates can achieve higher revenue. It's straightforward since more users are sampled, we can better set the price. As for different sampling rates under the same number of users, when there are more users, the revenue difference between different sampling rates becomes smaller. This means that all sampling rates have sampled enough user information to set a proper price. 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 excluded, and the utilization of the resources is low; when the sampling rate is low, the sampled users' information is not enough. As for the social welfare, the higher bias results in the higher social welfare, but the influence is less significant than the number of users.

7 Conclusion

In this work, we propose and study the Edge Pricing Game under Competition (EPGC). We consider the case where the cloud platform adopts fixed prices and investigate the best mechanism for the edge platform. We propose several truthful mechanisms that are restricted to setting prices for different unit resources. Though greedy mechanism have no theoretical guarantee on the revenue, it works well in a scarce market where the edge platform has relatively small computational resources. Random sampling mechanism has a 1/9 optimal revenue guarantee with probability approaching 1 when there is no single user can contribute a large fraction to the revenue. The performance is validated by our extensive experiments. One extension of this work is to consider the case that the cloud platform realizes the edge is a threat and will adjust the prices in response. Another direction is to design mechanisms that are not constrained to price for unit resources.

Acknowledgements

This work is supported partly by the National Key R&D Program of China 2018YFB0803400, NSFC Grants 61772489, 61751211, and Key Research Program of Frontier Sciences (CAS) No. QYZDY-SSW-JSC002. Haisheng Tan is the corresponding author (Email: hstan@ustc.edu.cn)

References

[1] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *computer*, vol. 50, no. 10, pp. 58–67, 2017.

[2] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computingâĂŤa key technology towards 5g," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[3] "Amazon web services (aws) - cloud computing services." `https://aws.amazon.com`.

[4] "App engine - build scalable web & mobile backends in any language — google cloud." `https://cloud.google.com/appengine/`.

[5] "Azure app service âĂŞ app hosting — microsoft azure." `https://azure.microsoft.com/en-us/services/app-service/`.

[6] S. Lehmann and P. Buxmann, "Pricing strategies of software vendors," *Business & Information Systems Engineering*, vol. 1, no. 6, p. 452, 2009.

[7] I. Ruiz-Agundez, Y. K. Penya, and P. G. Bringas, "A flexible accounting model for cloud computing," in *2011 Annual SRII Global Conference*, pp. 277–284, IEEE, 2011.

[8] M. Macías and J. Guitart, "A genetic model for pricing in cloud computing markets," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, pp. 113–118, ACM, 2011.

[9] B. Sharma, R. K. Thulasiram, P. Thulasiraman, S. K. Garg, and R. Buyya, "Pricing cloud compute commodities: A novel financial economic model," in *Proceedings of the 2012 12th IEEE/ACM international symposium on cluster, cloud and grid computing (ccgrid 2012)*, pp. 451–457, IEEE Computer Society, 2012.

[10] S. Zaman and D. Grosu, "Combinatorial auction-based allocation of virtual machine instances in clouds," *Journal of Parallel and Distributed Computing*, vol. 73, no. 4, pp. 495–508, 2013.

[11] L. Zhang, Z. Li, and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 433–441, IEEE, 2014.

[12] W. Shi, L. Zhang, C. Wu, Z. Li, and F. Lau, "An online auction framework for dynamic resource provisioning in cloud computing," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, pp. 71–83, ACM, 2014.

[13] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. Lau, "Online auctions in iaas clouds: Welfare and profit maximization with server costs," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, pp. 3–15, ACM, 2015.

[14] H. Zhang, H. Jiang, B. Li, F. Liu, A. V. Vasilakos, and J. Liu, "A framework for truthful online auctions in cloud computing with heterogeneous user demands," *IEEE Transactions on Computers*, vol. 65, no. 3, pp. 805–818, 2015.

[15] Z. Zhang, Z. Li, and C. Wu, "Optimal posted prices for online cloud resource allocation," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, p. 23, 2017.

[16] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2082–2091, 2017.

[17] T. Bahreini, H. Badri, and D. Grosu, "An envy-free auction mechanism for resource allocation in edge computing systems," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 313–322, IEEE, 2018.

[18] K. Xu, Y. Zhang, X. Shi, H. Wang, Y. Wang, and M. Shen, "Online combinatorial double auction for mobile cloud computing markets," in *2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, IEEE, 2014.

[19] W. Sun, J. Liu, Y. Yue, and H. Zhang, "Double auction-based resource allocation for mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4692–4701, 2018.

[20] "All pricing — compute engine documentation — google cloud." `https://cloud.google.com/compute/pricing#predefined`.

[21] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format+ schema," *Google Inc., White Paper*, pp. 1–14, 2011.

A The proof of Theorem 2

Proof. Consider the following example. There are n users with type $(d_i^1 = 1, d_i^2 = q^i, d_i^3 = 0, v_i = 1)$. The edge platform has capacity $(C_e^1 = n, C_e^2 = q^{n+1}, C_e^3 = 0)$. The cloud prices are $p_c^g = 0$.

The optimal mechanism should set prices $(1, 0, 0)$ and achieve revenue n. This is optimal since the users' value is n. Consider a mechanism sets uniform price $(p, p, p)$for all three resources. Suppose $p \in (\frac{1}{q^{k+1}+1}, \frac{1}{q^k+1}]$, then users $1 \le i \le k$ can afford the prices. The total payment would be

$$p \sum_{i=1,\dots,k} (1 + q^i) \le \frac{\sum_{i=1,\dots,k}(1 + q^i)}{q^k + 1} < 1 + \frac{2}{q}$$

So the mechanism with uniform prices has at most $\frac{1}{n} + \frac{2}{nq}$ fraction of the opitmal revenue. If we set q large enough, this number approaches $1/n$. □

B Proof of Theorem 4

Proof. When user i is chosen as samples, he has no incentive to misreport since his utility is doomed to be zero. When user i is not chosen as samples. The price is fixed already. What bias he reports only determines whether he would be in set T. Suppose he 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 becomes a member in A as a result. Then he will only get negative utility since his bias cannot afford the payment. If user i misreports and is not a member in T as a result. Then he will has zero utility.

In summary, misreport will not improve a user's utility in any case. So the mechanism is ex-post truthful. □

C The proof of Lemma 2

Proof. Since x_i are independent variable, and $x_i \in [0, h_i]$, by Hoeffding's inequality, we have

$$\begin{aligned} &\Pr(|\textstyle\sum_i x_i - E[\sum_i x_i]| \ge \frac{\sum_i h_i}{6}) \\ &\le 2\exp(-\frac{2(\frac{\sum_i h_i}{6})^2}{\sum_i h_i^2}) \end{aligned}$$

Thus, we have:

$$\begin{aligned} &\Pr(\textstyle\sum_i x_i \notin [\frac{\sum_i h_i}{3}, \frac{2\sum_i h_i}{3}]) \\ &\le 2\exp(-\frac{2(\frac{\sum_i h_i}{6})^2}{\sum_i h_i^2}) \\ &\le 2\exp(-\frac{2(\frac{\sum_i h_i}{6})^2}{\frac{\sum_i h_i}{\bar{h}}\bar{h}^2}) \\ &\le 2\exp(-\frac{\beta}{18}) \end{aligned}$$

The result is achieved. □