

## Research Article

# A Game-Based Scheme for Resource Purchasing and Pricing in MEC for Internet of Things

Yajing Leng <sup>1</sup>, Ming Wang <sup>1</sup>, Bowen Ma <sup>1</sup>, Ying Chen <sup>2</sup> and Jiwei Huang <sup>1</sup>

<sup>1</sup>Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum-Beijing, Beijing 102249, China

<sup>2</sup>Computer School, Beijing Information Science and Technology University, Beijing 100101, China

Correspondence should be addressed to Jiwei Huang; [huangjw@cup.edu.cn](mailto:huangjw@cup.edu.cn)

Received 6 September 2021; Revised 11 October 2021; Accepted 21 October 2021; Published 12 November 2021

Academic Editor: Xuyun Zhang

Copyright © 2021 Yajing Leng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile edge computing (MEC) is emerging as a promising paradigm to support the applications of Internet of Things (IoT). The edge servers bring computing resources to the edge of the network, so as to meet the delay requirements of the IoT devices' service requests. At the same time, the edge servers can gain profit by leasing computing resources to IoT users and realize the allocation of computing resources. How to determine a reasonable resource leasing price for the edge servers and how to determine the number of resource purchased by users with different needs is a challenging problem. In order to solve the problem, this paper proposes a game-based scheme for resource purchasing and pricing aiming at maximizing user utility and server profit. The interaction between users and the edge servers is modeled based on Stackelberg game theory. The properties of incentive compatibility and envy freeness are theoretically proved, and the existence of Stackelberg equilibrium is also proved. A game-based user resource purchasing algorithm called GURP and a game-based server resource pricing algorithm called GSRP are proposed. It is theoretically proven that solutions of the proposed algorithms satisfy the individual rationality property. Finally, simulation experiments are carried out, and the experimental results show that the GURP algorithm and the GSRP algorithm can quickly converge to the optimal solutions. Comparison experiments with the benchmark algorithms are also carried out, and the experimental results show that the GURP algorithm and the GSRP algorithm can maximize user utility and server profit.

## 1. Introduction

With the rapid development of Internet of Things (IoT) technology, various IoT devices such as smart phones and vehicles have been connected to the Internet [1, 2]. Service requests generated by IoT devices usually have strict requirements for computing resources and real-time processing [3]. Because IoT devices usually do not have enough computing resources [4], they usually offload service requests to the cloud for computing [5]. Generally speaking, large data processing centers or cloud servers are usually built in remote areas away from users. Therefore, when the service requests are offloaded to the cloud for computing, it will result in a lot of transmission costs and service delay. This is intolerable for IoT services that require high real-time performance.

To solve this problem, mobile edge computing (MEC) is proposed. MEC provides users with short-range cloud

computing services by deploying edge servers [6]. In MEC, users can offload service requests to the network edge for calculation [7]. The edge servers are close to users and have rich computing resources. Compared with the public cloud, the edge cloud is closer to the IoT devices, which can meet the requirements of IoT applications for low latency [8]. Because the user service request does not need to be transmitted to the remote cloud for calculation through the Internet, the transmission delay is reduced. In recent years, with the development of the IoT, a huge number of service requests have been offloaded to edge servers for computing [9]. Therefore, more and more edge cloud service providers came into being [10].

Although MEC can help provide resources for IoT applications, it faces unprecedented challenges. With the development of the IoT market, more and more different types of users will access the IoT networks [11]. Different users

have different purchasing needs for resources. Compared with the public cloud, there are some restrictions on the computing resources on the edge servers, which cannot meet the resource needs of all users. Therefore, how to reasonably allocate resources is a main challenge faced by MEC.

Reasonable pricing of resources can be used to solve the above problems. The servers price the provided computing resources and publish it to the users. Users choose appropriate resources to purchase according to the resource price of the servers and process the service request on the servers, so as to realize the reasonable allocation of resources. Therefore, the current resource pricing scheme in MEC needs to balance and meet the needs of different types of users.

In this work, we focus on resource purchasing under the condition of maximizing user utility and server profit. Its operation mechanism is as follows: the servers publish the resource leasing price, and then the users determine the number of resource purchasing. The servers obtain the resulting profit and repeatedly modify the leasing price in game. When the game equilibrium is reached, both the pricing of the servers and the resource purchasing of the users will be optimal.

Our contributions are summarized as follows:

- (i) We consider the scenario of an MEC system with multiple IoT device users and an edge server. Each user can purchase computing resources from the edge server and offload the service requests to the edge server for computing. We study the problem of resource purchasing and resource pricing from the perspective of users and servers and establish both the user utility function and the server profit function. The goal is to optimize both the user utility and the server profit together.
- (ii) We establish a Stackelberg game model to represent the interaction process of resource purchasing and resource pricing between multiple users and the server. The existence of Stackelberg equilibrium point is theoretically proved. It is also proved that the properties of incentive compatibility and envy freeness are satisfied. Then, we propose a game-based user resource purchasing algorithm (GURP) and a game-based server resource pricing algorithm (GSRP) which can obtain the optimal solution of Stackelberg equilibrium. We propose the theorem that the individual rationality property is satisfied.
- (iii) In order to verify the performance of our GURP and GSRP algorithms, we carry out simulation experiments. Experimental results show that the algorithms can eventually converge to the optimal solution. In addition, in terms of resource pricing and resource purchasing, two groups of comparison experiments with the benchmark algorithms are carried out. The results show that the GURP and GSRP algorithms can obtain the maximum user utility and server profit.

The remainder of this paper is organized as follows. We present the system model and relevant problem formulation in Section 2. We construct Stackelberg game model to analyze the interaction between users and servers and propose the GURP and GSRP algorithms in Section 3. We evaluate the performance of our GURP and GSRP algorithms in Section 4. The related works are reviewed in Section 5. The conclusion is given in Section 6.

## 2. System Model and Problem Formulation

*2.1. System Model.* An MEC system for the IoT considered in this paper consists of one edge server, denoted by  $S$ , and a set of users, denoted by  $U$ . Users can lease and purchase computing resources on the edge server and offload service requests to the edge server for computing. This can overcome the problem of insufficient local computing power of users. The edge server provides computing resource leasing services to users within their signal coverage in order to obtain profit. In this paper, we assume that the resources on the edge server can meet the needs of all users in its coverage.

We consider a game-based scene for resource purchasing and pricing in MEC shown in Figure 1. As mentioned earlier, at different times, users accessing the IoT have different needs and satisfactions with resources [12]. If the edge server always adopts a single resource pricing, it will have an impact on resource allocation and market economy.

Therefore, from the perspective of users and server, based on game theory, this paper determines the resource purchasing and resource pricing scheme that can optimize user utility and server profit.

*2.1.1. User Utility.* There are totally  $N$  users who propose the service requests, denoted by  $U = \{u_1, u_2, u_3, \dots, u_N\}$ . We assume that each user  $u_i$  proposes a service request. The service request of user  $u_i$  is specified as a tuple  $(C_i, T_i^m)$ .  $C_i$  represents the calculated size of  $u_i$  service request.  $T_i^m$  indicates the longest service request completion time acceptable to  $u_i$ .

We consider that all user's service requests must be transmitted before starting computing. Thus, the transmission time of the service request from user  $i$  to server is

$$T_i^s = \frac{C_i}{b}, \quad (1)$$

where  $b$  is the transfer rate.

According to the source price  $p$  published by the edge server, user  $i$  determines its resource purchasing strategy, which is denoted by  $a_i$ . The computing time of the service request from user  $i$  is

$$T_i^c = \frac{C_i \beta}{a_i f}, \quad (2)$$

where  $\beta$  represents the cycles per bit for computing one sample data of user and  $f$  represents the CPU frequency of a single resource in the edge server.

We define  $T_i$  as the actual completion time of the user's service request:

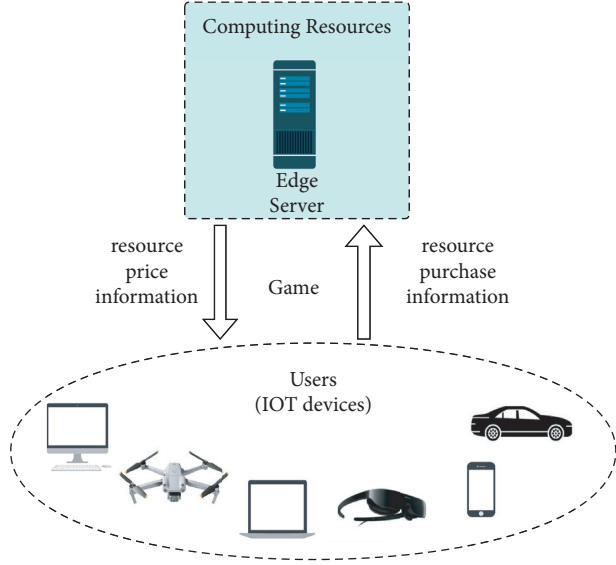


FIGURE 1: A game-based scene for resource purchasing and pricing in MEC.

$$\begin{aligned} T_i &= T_i^s + T_i^c \\ &= \frac{C_i}{b} + \frac{C_i\beta}{a_i f} \end{aligned} \quad (3)$$

The user utility of  $u_i$  is defined as

$$\begin{aligned} V_i &= \alpha_i \log\left(\frac{T_i^m}{T_i}\right) - pa_i \\ &= \alpha_i \log\left(\frac{T_i^m}{C_i/b + C_i\beta/a_i f}\right) - pa_i. \end{aligned} \quad (4)$$

where  $\alpha_i$  indicates user's  $u_i$  satisfaction with renting server resources. The higher the user satisfaction is, the more the users tend to purchase more server resources for service request calculation [13].  $pa_i$  represents the cost of user purchases edge server's resource.

**2.1.2. Edge Server Profit.** In addition to leasing computing resources to users, the edge server also needs to maintain computing resources. The edge server profit function can be defined as

$$M_s = p \sum_{i=1}^n a_i - q \sum_{i=1}^n a_i, \quad (5)$$

where  $q$  denotes the maintenance cost of the server to a single computing resource.

For the server, it only needs to maintain the resources leased to users. Other computing resources not leased to users will not incur maintenance costs.

The main notations and their definitions used in the following discussion are given in Table 1.

TABLE 1: Summary of key notations.

Notation	Definition
$u_i$	User $i$
$\alpha_i$	$u_i$ satisfaction with renting server resources
$T_i^m$	Longest service request completion time acceptable to $u_i$
$T_i$	Actual completion time of the $u_i$ service request
$a_i$	$u_i$ purchases the number of resources from the server
$\beta$	Cycles per bit for computing one sample data of user
$C_i$	Calculated size of $u_i$ service request
$p$	Price of a single server resource
$f$	CPU frequency of an edge server's single resource
$b$	Service request transfer rate
$q$	Maintenance cost of a server to a single resource
$V_i$	User $u_i$ utility
$M_s$	Edge server $S$ profit

**2.2. Problem Formulation.** We formulate the scheme for resource purchasing and pricing as a Stackelberg game. We divide the whole game process into two stages. In the first stage, the edge server determines its own resource pricing scheme. In the second stage, each user determines its resource purchasing strategy to maximize its own user utility. Therefore, in the process of this game, the edge server is a leader and users are followers. The strategy of the edge server is the source price  $p$  and the strategy of user is the number of resource purchasing, which is denoted by  $a_i$ . For the arbitrary pricing  $p$  of the edge server, user  $i$  will determine an optimal resource purchasing strategy to optimize its user utility, i.e.,

$$\begin{aligned} &\max V_i, \\ &s.t. V_i \geq 0, \\ &a_i \geq 0. \end{aligned} \quad (6)$$

The edge server will also update the pricing information according to users' resource purchasing strategies to pursue maximum profit, i.e.,

$$\begin{aligned} &\max M_s, \\ &s.t. M_s \geq 0. \end{aligned} \quad (7)$$

### 3. Game for Purchasing and Pricing Scheme

**3.1. User Utility Optimization.** User  $i$  needs to determine appropriate resource purchasing strategy according to the resource price of the edge server to maximize its own user utility. The problem is defined as follows:

$$\begin{aligned} Q_1 &= \max\{V_i\}, \\ V_i &= \alpha_i \log\left(\frac{T_i^m}{C_i/b + C_i\beta/a_i f}\right) - pa_i. \end{aligned} \quad (8)$$

The first derivative of  $V_i$  with regard to  $a_i$  is given by

$$\frac{dV_i}{da_i} = \frac{\alpha_i \beta b}{\ln 2 (a_i^2 f + a_i \beta b)} - p. \quad (9)$$

The second derivative of  $V_i$  with regard to  $a_i$  is given by

$$\frac{d^2 V_i}{da_i^2} = -\frac{1}{\ln 2} \frac{\alpha_i \beta b (2a_i f + \beta b)}{a_i^4 f^2 + a_i^2 \beta^2 b^2 + 2a_i^3 f \beta b} < 0. \quad (10)$$

Because the second derivative of  $V_i$  with regard to  $a_i$  is always negative, the function of  $V_i$  is a convex function.  $Q_1$  can be regarded as a convex optimization problem, and its optimal solution is

$$\frac{dV_i}{da_i} = 0, \quad (11)$$

i.e.,

$$a_i^* = \begin{cases} \frac{-\beta b + \sqrt{\beta^2 b^2 + 4f\alpha_i \beta b/p \ln 2}}{2f}, & V_i \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

When the resource unit price  $p$  is fixed, in order to obtain the maximum utility, the user purchases the number of resources as shown in (12). When the user utility value  $V_i \geq 0$ , the user chooses to purchase server resources and offload the service request to the server for calculation. Otherwise, the user will calculate the service request locally.

**Theorem 1** (incentive compatibility). *Users can truly report resource purchasing strategies. Users cannot obtain higher user utility by reporting false strategies.*

*Proof.* As proved earlier, user  $u_i$  determines the resource purchasing strategy  $a_i^*$  according to the resource unit price  $p$  formulated by the edge server.  $a_i^*$  is the unique maximizer of the user utility in equation (4). Then, its utility function satisfies

$$V_i(a_i^*) \geq V_i(a_i). \quad (13)$$

Therefore, user will not obtain better user utility through false reporting strategy. The user has no incentive to misreport its strategy. So, there exists incentive compatibility.  $\square$

**Theorem 2** (envy freeness). *The user always prefers its own purchased number of resources to that of others.*

*Proof.* In the system model proposed in this paper, users are independent of each other. All users can determine their resource purchasing strategies according to the price of edge server, so as to obtain the optimal user utility. The utility function of users only depends on their own resource purchasing strategies and resource unit price formulated by the edge server. Each user's resource purchasing strategy is optimal for itself. Therefore, users will not envy the strategies of other users.  $\square$

**3.2. Edge Server Profit Maximization.** *An edge server makes profit by leasing its computing resources. The server achieves*

*the goal of maximum profit by adjusting its resource unit price. The problem is defined as follows:*

$$Q_2 = \max\{M_s\}, \quad (14)$$

$$M_s = p \sum_{i=1}^n a_i - q \sum_{i=1}^n a_i,$$

where the value of  $a_i$  is determined by equation (12).

*The first derivative of  $M_s$  with regard to  $p$  is given by*

$$\frac{dM_s}{dp} = \sum_{i=1}^n \frac{\sqrt{b}(2fq\alpha_i + b \ln 2 \beta p^2 + 2fp\alpha_i)}{2f\sqrt{\ln 2} \sqrt{\beta} p^{3/2} \sqrt{\ln 2} b\beta p + 4f\alpha_i} - n \frac{b}{2f}. \quad (15)$$

*The second derivative of  $M_s$  with regard to  $p$  is given by*

$$\frac{d^2 M_s}{dp^2} = \sum_{i=1}^n -\frac{\sqrt{\beta} b ((2\beta b \alpha_i p \ln 2 + 6f\alpha_i^2)q + 2f\alpha_i^2 p)}{\sqrt{\ln 2} \sqrt{p} \sqrt{\beta} b p \ln 2 + 4f\alpha_i (\beta b \ln 2 p^3 + 4f\alpha_i p^2)} < 0. \quad (16)$$

Because the second derivative of  $M_s$  with respect to  $p$  is always negative, the function of  $M_s$  is a convex function.  $p \rightarrow 0, M_s < 0$ ;  $p \rightarrow \infty, M_s = 0$ . Thus,  $Q_2$  can be regarded as a convex optimization problem, and it has a unique optimal solution  $p^*$ . The optimal solution  $p^*$  is related to the satisfaction of leasing resources of each user ( $\alpha_i$ ).

**3.3. Stackelberg Equilibrium.** *For users and the edge server, in the game model, the existence of Stackelberg equilibrium can be proved by the existence of optimal solutions for problems  $Q_1$  and  $Q_2$ . This not only ensures that the edge server can get the optimal profit but also ensures that users can get the optimal utility.*

**Theorem 3.** *For the edge server, there is an optimal resource price  $p^*$ , which makes the server profit optimal.  $u_i$  has an optimal resource purchasing strategy  $a_i^*$ , which makes the user utility optimal. Then, it can be explained that the game model has a Stackelberg equilibrium, i.e.,*

$$M_s(p^*) \geq M_s(p), \quad (17)$$

$$V_i(a_i^*) \geq V_i(a_i).$$

*Proof.* It can be obtained from formula (16) that the edge server profit  $M_s$  is a convex function with regard to resource price  $p$ . Therefore, the edge server can get an optimal resource pricing strategy, so as to maximize the server profit. For a certain resource price, according to (12), user can make an optimal source purchasing strategy to maximize personal utility. Therefore, there is Stackelberg equilibrium in the game model.  $\square$

### 3.4. Algorithm Design

**3.4.1. Game-Based User Resource Purchasing Algorithm.** We propose the game-based user resource purchasing (GURP) algorithm as shown in Algorithm 1. For each user, in each game with the server, they will first accept the

resource price  $p$  information published by the server. The user determines the resource purchasing strategy that maximizes the user's utility according to formula (12). Due to individual rationality, each user will judge whether its utility value is greater than 0. If the utility value is less than 0, the user will give up purchasing computing resources. On the contrary, the user determines the resource purchasing strategy and reports the strategy information to the server.

The specific process is as follows. In line 2, we initially set the user utility values of all users to 0. In lines 4–14, each user sets resource purchasing strategy according to (12). Finally, user will make a resource purchasing strategy to maximize its user utility and report the resource purchasing strategy to the edge server.

**3.4.2. Game-Based Server Resource Pricing Algorithm.** We propose the game-based server resource pricing (GSRP) algorithm as shown in Algorithm 2. For the server, at the beginning of the game, set a small resource pricing information and publish the information to the user. Then, the server accepts the purchase information of users and calculates its own profit. Next, the server sets an appropriate resource price update step. It continuously updates the resource pricing information and publishes it to users and accepts the user's purchase information and calculates the profit. In the iterative process of game between the server and the users, the server's pricing strategy will converge to the price that maximizes the profit of server.

The specific process is as follows. In lines 1–5, we initialize the variable values in the Algorithm 2. In lines 6–12, the edge server will constantly update the resource price to maximize the server profit and record the optimal resource price. Finally, the edge server will get the resource price which can maximize the profit.

Both Algorithm 1 and 2 have high computational efficiency. For GURP, as shown in Algorithm 1, for line 4, because of  $n$  users participating, it needs to cycle  $n$  times for calculation. Therefore, the computational complexity of Algorithm 1 is  $\mathcal{O}(N)$ . For GSRP, as shown in Algorithm 2, for lines 6–12, the number of iterations for the convergence of server profit is limited. We use  $M$  to represent the number of iterations, so its computational complexity is  $\mathcal{O}(M)$ . Then, from line 10, the computational complexity is  $\mathcal{O}(N)$ . Therefore, the computational complexity of Algorithm 2 is  $\mathcal{O}(MN)$ .

**Theorem 4.** Both GURP and GSRP mechanisms satisfy individual rationality.

*Proof.* Individual rationality means that no one will suffer from participating in the sale mechanism. For resource purchasers (users), in GURP, as shown in Algorithm 1, for line 6, users will purchase resources on the premise that their user utility is greater than 0. For resource seller (edge server), in GSRP, as shown in Algorithm 2, for line 6, the resource price set by the edge server must make its profit greater than 0. Therefore, both GURP and GSRP mechanisms satisfy individual rationality.  $\square$

## 4. Performance Evaluation

**4.1. Setup.** We conduct simulation experiments and use simulation data to verify the game algorithm proposed in this paper. In the experimental scenario, there are six users and an edge server. We set the maintenance cost of a server to a single resource  $q$  as 1 [14] and set the CPU frequency of an edge server's single resource  $f$  as 4 GHz [15]. The satisfaction of  $u_i$  is set to 50–100 [16].

To be specific, the main parameters involved in this experiment are shown in Table 2.

**4.2. Parametric Analysis.** The first set of experiments is to investigate the change of edge server profit in the game. From the result shown in Figure 2, we can know that the profit of the server will converge to the equilibrium point of the game with the increase of the number of games. The convergence rate is related to the update step of  $p$ . When step  $\Delta p$  is small, the server profit will have to go through multiple rounds of iteration to reach the convergence point. When step  $\Delta p$  is big, the server profit value may miss the convergence point.

The game iteration between users and the server takes some time and energy. Too many iterations may cause users to give up resource purchasing because they cannot stand the consumption of time and energy. Too few iterations may cause the server to miss the optimal pricing strategy and damage the profit of the service provider. Therefore, in an actual scenario, the update step size of the server resource pricing strategy will have an impact on the final profit of the server. The server needs to formulate a reasonable resource unit price update step.

Figure 3 shows the change of user's utility value with the number of iterations. We study and analyze the change of utility value of six users with different satisfaction and different service request sizes. The user's satisfaction relationship is  $\alpha_1 < \alpha_2 < \alpha_3 < \alpha_4 < \alpha_5 < \alpha_6$ . At the beginning, the server makes the price of resources very low, so users will have high user utility. With the progress of the game, the server will gradually formulate optimal resource price, so user's utility will gradually decrease and reach the equilibrium point of the game. The higher the user's satisfaction with the leased resources is, the more the user tends to purchase more edge computing resources to obtain greater user utility. The higher the satisfaction of users, the higher the user's utility after reaching the equilibrium point of the game.

We specifically study the impact of satisfaction on user's utility and server's profit. From the result shown in Figure 4, for users, users with high satisfaction tend to buy more computing resources at the edge server, so as to obtain greater user's utility. For the server, simultaneously, providing more resources to the group of users with high satisfaction can bring in higher profits.

Finally, Figure 5 illustrates the impact of service request transfer rate and CPU frequency on server's profit. It is shown that as the service request transfer rate and CPU frequency increase, the server's profit becomes higher. The server provides high transmission rate channel and high

```

Input:  $p$ 
Output: optimal  $a_i^*$  for each  $u_i$ 
(1) for each  $u_i$  in  $U$  do
(2)    $V_i = 0$ ;
(3) end
(4) for each  $u_i$  in  $U$  do
(5)    $a_i = -\beta b + \sqrt{\beta^2 b^2 + 4f\alpha_i\beta b/p \ln 2/2f}$ 
(6)   if  $V_i > 0$  then
(7)      $a_i^* = -\beta b + \sqrt{\beta^2 b^2 + 4f\alpha_i\beta b/p \ln 2/2f}$ ;
(8)      $V_i = \alpha_i \log(T_i^m/C_i/b + C_i\beta/a_i^* f) - p a_i^*$ ;
(9)   else
(10)     $a_i^* = 0$ ;
(11)     $V_i = 0$ ;
(12)  end
(13) end
(14) return  $a_i^*$  for each  $u_i$ ;

```

ALGORITHM 1: Game-based user resource purchase algorithm (GURP).

```

Input:  $n, \sum_{i=1}^n a_i$ 
Output: optimal price  $p^*$ 
(1)  $M_s^* = 0$ ;
(2)  $p^* = 0$ ;
(3)  $q = \text{maintenance cost}$ ;
(4)  $p = \text{initial price}$ ;
(5)  $M_s = p \sum_{i=1}^n a_i - q \sum_{i=1}^n a_i$ ;
(6) while  $M_s > M_s^*$  do
(7)    $M_s^* = M_s$ ;
(8)    $p^* = p$ ;
(9)    $p = p + \Delta p$ ; (update  $p$ )
(10)  GURP ( $p$ );
(11)   $M_s = p \sum_{i=1}^n a_i - q \sum_{i=1}^n a_i$ ;
(12) end
(13) return  $p^*$ ;

```

ALGORITHM 2: Game-based server resource pricing algorithm (GSRP).

TABLE 2: Experiment setup.

Parameters	Value
$\alpha_i$ ( $u_i$ satisfaction)	[50, 100]
$C_i$ (calculated size of $u_i$ service request)	[60, 80] bit
$f$ (CPU frequency of an edge server's single resource)	4 GHz
$\beta$ (cycles per bit for computing one sample data of user)	20 cycles/bit
$b$ (transfer rate)	$1 \times 10^8$ bit/s

computing power resources, which can attract users to purchase more computing resources, so as to improve the profit of the server.

**4.3. Comparison Experiment.** In this part, aiming at the problem of resource pricing and resource purchasing, we compare and analyze many different resource pricing and resource purchasing algorithms and further evaluate the performance of the game algorithm (GURP and GSRP) proposed in this paper.

**4.3.1. Resource Pricing.** The server leases resources to users, determines the resource leasing price, and obtains profit by charging users a fee. For server's resource pricing, we compare three pricing strategies:

- (i) Random resource pricing: the server randomly makes a resource leasing price.
- (ii) Historical optimal resource pricing: the server queries the historical optimal resource pricing and takes it as the current resource pricing scheme.

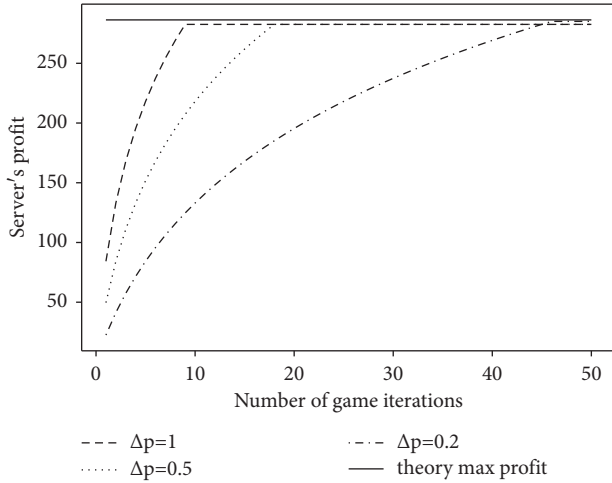


FIGURE 2: The profit of the server in game process.

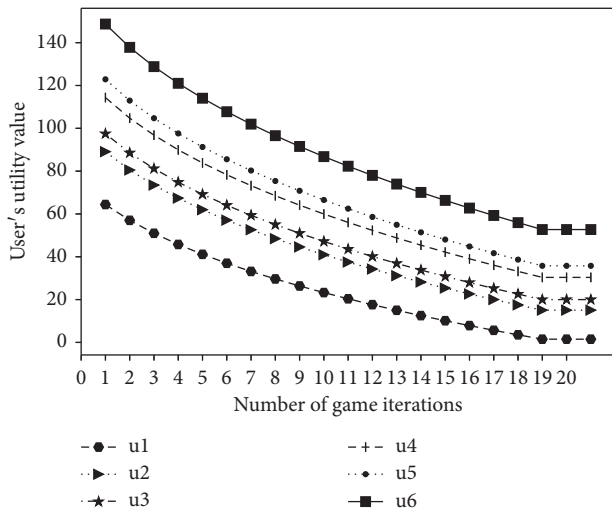


FIGURE 3: The utility values of users in game process.

- (iii) Game-based server resource pricing (GSRP): in the process of game with users, the server continuously updates the price until the optimal resource pricing scheme is given.

Figure 6 shows the server's profit with different resource pricing strategies. The experimental result shows that the profit of the server is different for the user groups with different satisfaction. With the increase of user satisfaction, the profit of server also increases. The game-based pricing algorithm proposed in this paper is superior to the other two kinds of pricing algorithms in maximizing server profit. The GSRP algorithm can find the most suitable resource price for the current user group and obtain the maximum profit in the process of game with users.

**4.3.2. Resource Purchasing.** In order to maximize user utility, users purchase appropriate edge computing resources to calculate service requests. For user's resource purchasing, we compare three resource purchasing strategies:

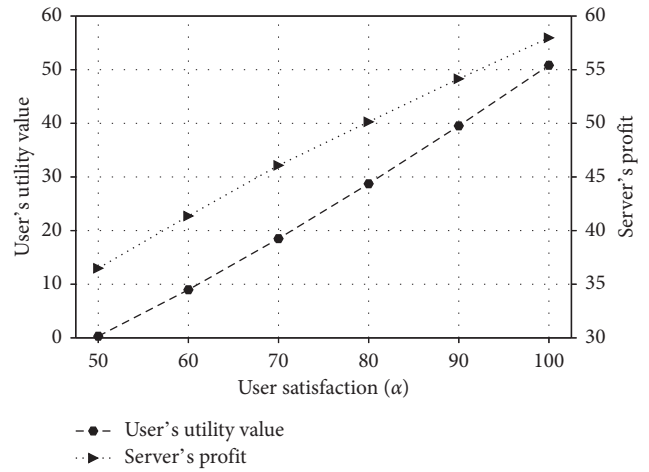


FIGURE 4: Impact of user's satisfaction.

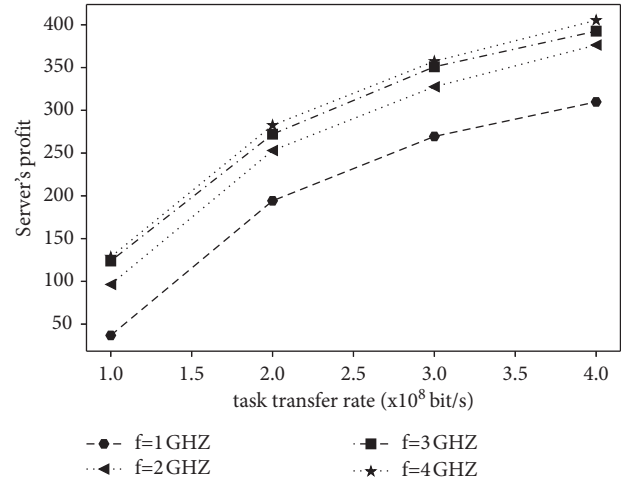


FIGURE 5: Impact of calculation frequency and transfer rate.

- (i) Random resource purchasing: user randomly purchases a certain number of resources on the edge server.
- (ii) Fixed resource purchasing: user purchases a fixed number of resources on the edge server according to the size of its service request. User with larger service request will purchase more edge server resources for calculation.
- (iii) Game-based user resource purchasing (GURP): in the process of game with server, user determines the resource purchasing strategy to maximize its user's utility according to the resource pricing of the edge server.

Figure 7 shows the utility values of users with different resource purchasing strategies. The experimental result shows that the user's utility is different for the users with different satisfaction. The more satisfied the users are, the more inclined they are to participate in the resource purchasing market, so as to obtain greater utility. The game-based resource purchasing algorithm proposed in this paper is superior to the other two kinds of resource purchasing

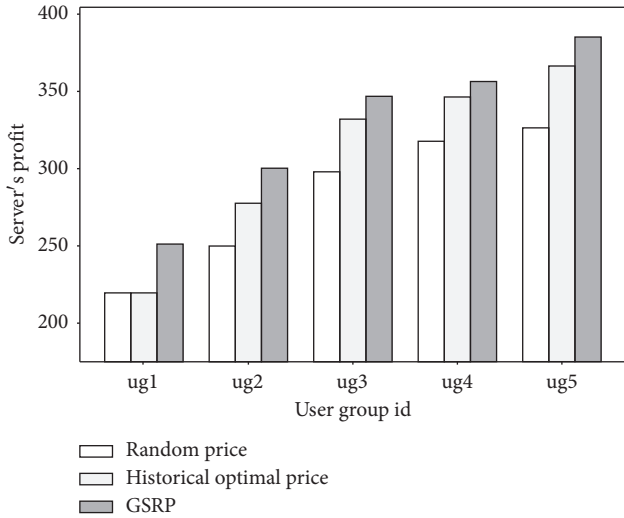


FIGURE 6: The profit of the server with different pricing strategies.

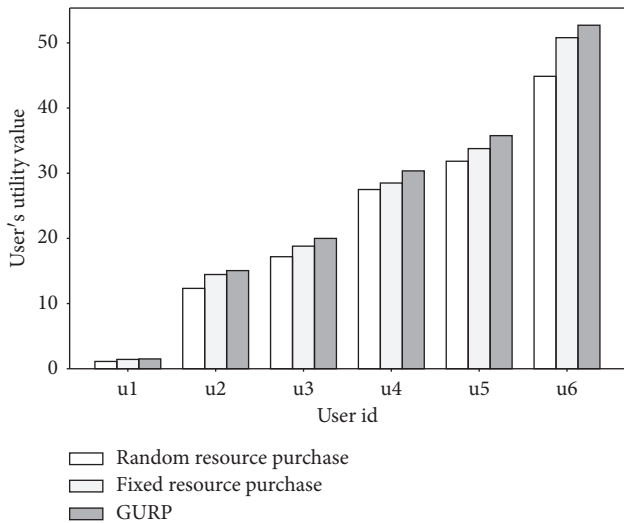


FIGURE 7: The utility values of users with different resource purchasing strategies.

algorithms in maximizing user's utility. The GURP algorithm can determine the best resource purchasing strategy according to the real-time resource price and obtain the optimal user utility.

## 5. Related Works

In recent years, with the development of the IoT, many research studies focus on its resource allocation and resource pricing problems [17].

In [18], the authors adopt the machine learning method for the first time and obtain a market model by using big data. Through the model, servers can get an optimal resource pricing scheme. Siew et al. [19] mainly studied the resource allocation in MEC. From the perspective of maximizing servers' profits, the authors designed two dynamic resource pricing mechanisms for resource allocation. In [20], from

the perspective of users, users can independently select server resources by judging the acceptable service request price and time cost to realize resource allocation. In [21], the authors introduced a Petri net and proposed a resource allocation strategy based on pricing time. In [22], the authors studied a fog calculation scene. By judging the priority of users requests, the servers formulate relevant pricing schemes and resource allocation schemes.

Due to the resource leasing and resource purchasing behavior between servers and users, some studies focus on the economic problems in MEC. In [23], in order to solve the resource allocation problem in MEC, the authors proposed two dynamic pricing double auction algorithms. From the perspective of maximizing social welfare, the authors in [24] introduced a broker between users and servers to manage market purchasing and pricing behavior and proposed an iterative bilateral auction scheme. In [25], Stackelberg game was applied to the task offloading for mobile blockchain. The authors proposed a dual auction game mechanism to obtain the optimal resource price and equipment resource demand.

The Stackelberg game theory is widely used to solve the problem of resource management. The authors in [16] studied a scenario of Internet of vehicles, and the real-time pricing problem of computing resources in Internet of vehicles was solved by using Stackelberg game. In [26], the authors studied the allocation of computing resources in multiple edge clouds and ToT devices and proposed a computing offload mechanism based on two-stage Stackelberg game. Zhang et al. [27] proposed a distributed algorithm for resource allocation based on Stackelberg game.

However, few studies pay attention to the problem that users actively determine the number of resources to purchase according to the real-time resource pricing of servers in MEC. This paper studies the resource purchasing and resource pricing scheme in MEC from the perspective of Stackelberg game theory. The Stackelberg game theory is used to model the interaction between edge servers and users, and the existence of Stackelberg equilibrium is proved. In addition, the resource purchasing and resource pricing algorithms based on game theory are proposed.

## 6. Conclusion

In this paper, we investigate a game-based scheme for resource purchasing and pricing in MEC for IoT. Based on Stackelberg game, the server and users can update their resource pricing strategy and their resource purchasing strategies continuously during the game. The models of optimal user utility and server profit are given. We propose a game-based scheme for resource purchasing and pricing, and the existence of Stackelberg equilibrium is proved. Finally, the algorithm is evaluated by simulation experiments. The experiment results demonstrate that user utility value and edge server profit obtained by this algorithm are better than other basic resource purchasing and resource pricing algorithms. For our future work, we will consider unloading part of the user's service request to the edge server for calculation. Service requests will be calculated in parallel on local and edge servers. Users can maximize their user utility



values by determining the size of offloading service requests and the number of resource purchasing.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This study was supported by the National Natural Science Foundation of China (nos. 61972414, 61902029, and 61973161), Beijing Nova Program (no. Z201100006820082), Beijing Natural Science Foundation (no. 4202066), Fundamental Research Funds for Central Universities (no. 2462018YJRC040), Excellent Talents Projects of Beijing (no. 9111923401), and Scientific Research Project of Beijing Municipal Education Commission (no. KM202011232015).

## References

- [1] Y. Liu, D. Li, S. Wan et al., "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, 2021.
- [2] J. Mabrouki, M. Azrou, G. Fattah, D. Dhiba, and S. E. Hajjaji, "Intelligent monitoring system for biogas detection based on the internet of things: mohammedia, Morocco city landfill case," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 10–17, 2021.
- [3] S. Chen, L. Zhang, Y. Tang et al., "Indoor temperature monitoring using wireless sensor networks: a SMAC application in smart cities," *Sustainable Cities and Society*, vol. 61, Article ID 102333, 2020.
- [4] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.
- [5] W. Zhang, X. Chen, and J. Jiang, "A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems," *Tsinghua Science and Technology*, vol. 26, no. 1, pp. 95–111, 2020.
- [6] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "Energy efficient dynamic offloading in mobile edge computing for Internet of Things," *IEEE Transactions on Cloud Computing*, vol. 9, 2019.
- [7] R. Bi, Q. Liu, J. Ren, and G. Tan, "Utility aware offloading for mobile-edge computing," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 239–250, 2020.
- [8] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 2019, Article ID 292369, 2019.
- [9] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4925–4934, 2020.
- [10] D. Kim, J. Son, D. Seo, Y. Kim, H. Kim, and J. T. Seo, "A novel transparent and auditable fog-assisted cloud storage with compensation mechanism," *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 28–43, 2019.
- [11] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.
- [12] Y. N. Malek, M. Najib, M. Bakhouya, and M. Essaïdi, "Multivariate deep learning approach for electric vehicle speed forecasting," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 56–64, 2021.
- [13] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, vol. 2020, Article ID 2969489, 2020.
- [14] X. Yu and L. Tang, "Competition and cooperation between edge and remote clouds: a stackelberg game approach," in *Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pp. 1919–1923, IEEE, Chengdu, China, December 2018.
- [15] T. Mahn, H. Al-Shatri, and A. Klein, "Distributed algorithm for energy efficient joint cloud and edge computing with splittable tasks," in *Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–7, IEEE, Marrakesh, Morocco, April 2019.
- [16] C. Tang, C. Zhu, H. Wu, X. Wei, Q. Li, and J. J. Rodrigues, "A game theoretical pricing scheme for vehicles in vehicular edge computing," in *Proceedings of the 2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 17–22, IEEE, Tokyo, Japan, December 2020.
- [17] J. Huang, S. Li, and Y. Chen, "Revenue-optimal task scheduling and resource management for IoT batch jobs in mobile edge computing," *Peer-to-Peer Networking and Applications*, vol. 13, no. 5, pp. 1776–1787, 2020.
- [18] D. Niyato, M. A. Alsheikh, P. Wang, D. I. Kim, and Z. Han, "Market model and optimal pricing scheme of big data and internet of things (IoT)," in *Proceedings of the 2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Kuala Lumpur, Malaysia, May 2016.
- [19] M. Siew, D. Cai, L. Li, and T. Q. S. Quek, "Dynamic pricing for resource-quota sharing in multi-access edge computing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2901–2912, 2020.
- [20] L. Ni, J. Zhang, and J. Yu, "Priced timed Petri nets based resource allocation strategy for fog computing," in *Proceedings of the 2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*, pp. 39–44, IEEE, Beijing, China, October 2016.
- [21] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in fog computing based on priced timed petri nets," *Ieee Internet of Things Journal*, vol. 4, no. 5, pp. 1216–1228, 2017.
- [22] A. Sutagundar and S. B. Shahapur, "Development of fog based dynamic resource allocation and pricing model in IoT," in *Proceedings of the 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 349–354, IEEE, Bangalore, India, August 2018.
- [23] 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.

- [24] Z. Li, Z. Yang, and S. Xie, "Computing resource trading for edge-cloud-assisted internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3661–3669, 2019.
- [25] S. Guo, Y. Dai, S. Guo, X. Qiu, and F. Qi, "Blockchain meets edge computing: stackelberg game and double auction based task offloading for mobile blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5549–5561, 2020.
- [26] F. Li, H. Yao, J. Du, C. Jiang, and Y. Qian, "Stackelberg game-based computation offloading in social and cognitive industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5444–5455, 2019.
- [27] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: a joint optimization approach combining Stackelberg game and matching," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1204–1215, 2017.