

RESOURCE PRICING MODEL BASED ON DOUBLE AUCTION FOR THE CLOUDLET FEDERATION

HENGZHOU YE, YIHUI CHEN AND XINXIAO LI*

Guangxi Key Laboratory of Embedded Technology and Intelligent System
Guilin University of Technology

No. 319, Yanshan Street, Yanshan District, Guilin 541006, P. R. China
{2002018; 2120200935}@glut.edu.cn; *Corresponding author: renpet@glut.edu.cn

Received August 2022; revised November 2022

ABSTRACT. *The cloudlet federation is a new computing architecture which reduces the cost of cloudlet deployment and management by sharing cloudlet resources among cloudlet service providers. Renting resources is an important means for the cloudlet federation to improve resource utilization rate. However, the cloudlet federation has the problem of how to formulate a reasonable and effective resource pricing mechanism. Therefore, we construct a resource pricing model based on double auction, aiming at maximizing the benefits of the cloudlet federation. The reasonable dynamic pricing strategies for resource providers and resource demanders are formulated from different perspectives: the asking price of the resource provider considers the resource utilization rate, and the bid of the resource demander considers the task deadline, and the price adjustment strategy is used to conduct multiple rounds of auctions to improve the success rate of task auction. Simulation results show that the proposed dynamic resource pricing algorithm has better performance than the other two fixed resource pricing algorithms in terms of the total benefits of the cloudlet federation, resource utilization rate and auction task success rate.*

Keywords: Cloudlet federation, Resource pricing, Double auction, Multi-round auction

1. Introduction. With the continuous development of the Internet of Things and mobile device technologies, applications such as autonomous driving, online shopping and online chat continue to emerge. A large amount of complex and heterogeneous data leads to network congestion, long response delays, and difficulty in satisfying user experience, so mobile edge computing (MEC) [1] is proposed to solve the above problems. Mobile edge computing is usually deployed on the edge of the network to provide computing, storage and data transmission services for terminal devices, featuring high bandwidth and low latency. Cloudlet [2] is an important way to realize mobile edge computing, which is a stable, reliable and resource-rich small server cluster that can provide data processing, storage and other services for nearby applications. Edge infrastructure providers (EIPs) are more inclined to user-specific needs than cloud providers, thus establishing some private edge computing environments, but also resulting in high deployment and maintenance costs. EIPs are independent highly, do not share resources, resulting in low resource utilization rate. In response to this problem, the latest research proposes a new paradigm of edge computing – edge federation [3], and provides intelligent resource scheduling and high-performance optimization solutions in the cloudlet federation system. As shown in Figure 1, the cloudlet federation forms a resource pool to serve cloudlet service providers (CSPs) and users by integrating various EIPs and cloud resources to achieve the goals of improving resource utilization rate and reducing service costs. Although CSPs have

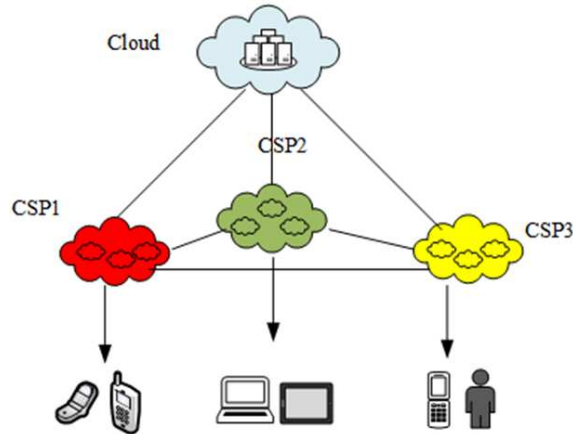


FIGURE 1. Cloudlet federation framework

formed federation, there are still some CSPs with insufficient resources to provide services for user tasks, while some CSPs are in a state of idle resources and low resource utilization rate. Therefore, renting resources is an important means of realization, that is, CSPs which require cloudlet resources rent cloudlet resources from other CSPs at a certain price. However, when renting resources, the cloudlet federation has to formulate a reasonable and effective resource pricing mechanism. What kind of pricing mechanism should CSPs use to rent the cloudlet resources of other CSPs or lease their own cloudlet resources to other CSPs is the focus of our attention.

Resource pricing based on auctions [4,5] is a very typical pricing method, and good research results have been obtained in many engineering fields, such as parking spaces [6], multi-sensors [7], and sharing energy storage [8]. Because the auction-based pricing mechanism needs less information, is easy to implement and has a low cost, the paper adopts the auction method to realize the pricing of cloudlet resources. The auction method is divided into one-way auction [9] and double auction [10]. The mode of one-way auction is usually “1 : N”, and the number of one party in the transaction is only one, and the other party is multiple. One-way auctions are easy to make one auctioneer occupy an advantageous position to control the market initiative and obtain higher profits easily, but it is also easy to damage the benefits of the other party. However, the two-way auction adopts the “N : N” mode, that is, there is more than one party to the transaction, which makes both parties to the transaction equal, the market relatively fair. These can improve the auction efficiency, and ensure the benefits of both parties. Therefore, it is more in our benefits to use a double auction to achieve resource pricing in the cloudlet federation.

At present, there are few researches on the resource leasing of the cloudlet federation. Baghban et al. [11] studied the end-to-end delay problem through resource sharing between edge nodes. Moon and Lim [12] proposed a task division method in edge consortium to optimize load and improve resource utilization rate. In terms of resource leasing of the cloudlet federation, [13,14] only proposed leasing optimization schemes from the perspective of the benefits of resource providers. In [15], edge resource demanders rent resources from other resource providers through a resource pricing mechanism based on the market supply and demand function to maximize the benefits of resource providers. However, these papers only consider the benefits of resource providers, ignoring the benefits of resource demanders. This phenomenon leads to excessive quotation and vicious competition, which is not conducive to the good prospects of the development of the cloudlet federation.

Therefore, in order to provide a good resource pricing mechanism in the cloudlet federation, this paper starts from the overall benefit of the cloudlet federation: the resource provider's asking price strategy considers the resource utilization rate, and the resource demander's bidding strategy considers the task deadline. We establish a dynamic resource pricing model based on double auction with the goal of maximizing the total benefits of the cloudlet federation. The model can adjust the quotations of both parties dynamically and flexibly, design a price adjustment strategy to conduct multiple rounds of auctions, and propose a resource pricing algorithm (RPA) which is compared with the fixed pricing algorithm in [16,17], to highlight the rationality of RPA, which provides a good resource pricing mechanism when renting resources.

The main contributions of this paper are summarized as follows.

1) In order to ensure the benefits of both parties, we propose a framework based on double auction and design a dynamic pricing strategy. The resource provider's asking price strategy considers resource utilization rate, and the resource demander's bidding strategy considers task deadlines. This paper formulates a price adjustment strategy to conduct multiple rounds of auctions to improve task auction success rate.

2) From an overall perspective, we design a dynamic resource pricing process based on double auction, and design the algorithm RPA to select the cloudlet with the lowest asking price for each task in turn.

3) Through a large number of experiments, we compare the performance of the dynamic resource pricing algorithm we designed and the other two pricing algorithms from the perspectives of the total benefits of the cloudlet federation, resource utilization rate and task auction success rate.

The rest of the paper is as follows: Section 2 describes the related work, and Section 3 builds a resource pricing model for the cloudlet federation. Section 3 also introduces the dynamic resource pricing algorithm based on double auction. Section 4 presents the experimental results. Finally, Section 5 discusses our conclusions.

2. Related Work. The cloudlet federation is essential to integrate and share the cloudlet resources of multiple cloudlet service providers, and solve the problem of insufficient resources of a single cloudlet service provider by renting resources. Huang et al. [15] proposed a resource lease revenue model based on pricing mechanism in the cloudlet federation, so that resource providers can rent resources from other resource providers with reasonable prices. Luo and Qiu [18] calculated how many resources need to be reserved or leased based on queuing theory and probability analysis from the perspective of resource providers to maximize the profits of service providers. Awada and Zhang [19] proposed an intelligent scheduling strategy and considered the dependence of tasks and heterogeneous resource requirements. Multiple tasks were concentrated on a node to improve resource utilization. Lin et al. [20] proposed proxy based joint authentication protocols for cloud and edge scenarios to shorten the delayed authentication time. Kelaidonis et al. [21] proposed a federation based edge Cloud-IoT architecture to manage cross domain services of different Cloud-IoT resources and reduce service delay. Randriamasinoro et al. [22] proposed a resource allocation optimization model in the cloudlet federation according to the quality of service requirements, and optimized the allocation of tasks on each computing node, aiming to reduce the cost of resource providers. Most of the existing papers focus on the time delay, resource utilization rate and the benefits of resource providers of the edge federation. However, the research based on the total benefits of the system under the cloudlet federation is still blank.

In recent years, the current research on resource pricing at home and abroad is mainly divided into static pricing and dynamic pricing. Many companies have adopted static

resource pricing methods, such as Amazon EC2, Microsoft Windows Azure and Google's Engine platform, which adopts a fixed-proportion pricing method. Teng and Magoulès [23] allocated resources according to the proportion of user bids, and the amount of resources provided by resource providers increases with the increase of bids. Zhang et al. [17] studied the auction matching problem between multiple edge service providers and multiple user equipment, firstly pricing according to user needs, and then adjusting the price according to a fixed ratio, which is better than other algorithms in terms of utility and the number of auction winning tasks.

Although the static pricing strategy is easy to understand and use, the model is fixed, single and inflexible. It is mainly based on the benefits of resource providers, ignoring the benefits of resource demanders, and cannot dynamically adjust prices according to resource supply and demand, resulting in lower auction efficiency and resource utilization rate. Therefore, the dynamic pricing strategy is more and more favored by scholars. Hu and Liu [24] proposed a dynamic resource pricing model based on double auction. Both parties adjust the price in real time according to the response time and resource utilization rate. Sun et al. [25] proposed a heuristic cloud resource allocation algorithm based on a continuous double combinatorial auction based on Nash equilibrium characteristics. The resource pricing considers the load, and the user pricing considers the remaining resources of the system. Dibaj et al. [26] proposed a dynamic online combination double auction mechanism based on cloud priority, and the pricing mechanism considers constraints such as time evanescence. Yadav and Kumar [27] discussed the pricing model of DaaS based on various cloudlet frameworks, and proposed a pricing mechanism that considers task granularity, cost, and utility factors. The improved algorithm is incentive compatible. Mihailescu and Teo [28] proposed a dynamic pricing strategy to meet user requests for multiple resource types and adjust prices according to resource demand. Robertus et al. [29] proposed to introduce cloud computing into the research of the Internet of Things to optimize the pricing of the healthcare industry. Xu and Palanisamy [30] proposed a contract based joint cloud resource sharing mechanism, and resource providers adopted cost aware job scheduling and supply dynamic algorithms to maximize their benefits under the condition of meeting the response time of tasks.

However, the above studies are more focused on maximizing the benefits of resource providers. Under the premise of comprehensively considering the factors of resource utilization rate and task deadlines, from the perspective of both parties as a whole to research the problem of resource pricing, a dynamic pricing model aiming at maximizing the total benefits of the cloudlet federation is proposed. The model can adjust the prices of both parties flexibly to win more total benefits of the cloudlet federation.

3. System Model. The resource pricing system based on double auction consists of N resource demanders, M resource providers and one auctioneer *RIM* (resource information manager). $R = \{r_1, r_2, \dots, r_N\}$ represents a collection of resource demanders. $P = \{p_1, p_2, \dots, p_M\}$ represents a collection of resource provider *RIM* acts as a third-party auctioneer and is responsible for collecting bidding information from CSPs and completing resource rental allocation. During the auction process, trading parties shall not conspire, and the bidding information shall be sent in a sealed form. The auctioneer shall remain impartial and public, serve both parties free of charge, and the bidding information shall be kept confidential. Tasks are independent of each other. One task can only rent the resources of one cloudlet, and one cloudlet can rent resources for multiple tasks. When the resources of a cloudlet are leased to a task, the resource amount of its resource package needs to be updated. The major notations used in the paper are listed in Table 1.

TABLE 1. Main notations

Notation	Description
R	A collection of resource demanders
P	A collection of resource providers
T_n	An auction time period
r_i	The i -th resource demander
T_i	A collection of task package of r_i
$T_{i,j}$	The j -th user task of T_i
p_k	The k -th resource provider
S_k	A collection of cloudlet resource package of p_k
C_k	A collection of cloudlets of p_k
$c_{k,q}$	The k -th cloudlet of p_k
n_i	The number of user tasks of T_i
m_k	The number of cloudlets of p_k
$Rent_{i,j}^{k,q}$	A binary identifier, indicating whether $c_{k,q}$ leases resources to $T_{i,j}$
$price_{i,j}^{k,q}$	The transaction price $T_{i,j}$ and $c_{k,q}$
$ask_{k,q}^t$	The asking price of the unit computing resources of $c_{k,q}$ at time t
$bid_{i,j}$	The bid of the unit computing resource of $T_{i,j}$
PU_k	The benefits of the resource provider p_k
RU_i	The benefits of the resource demander r_i

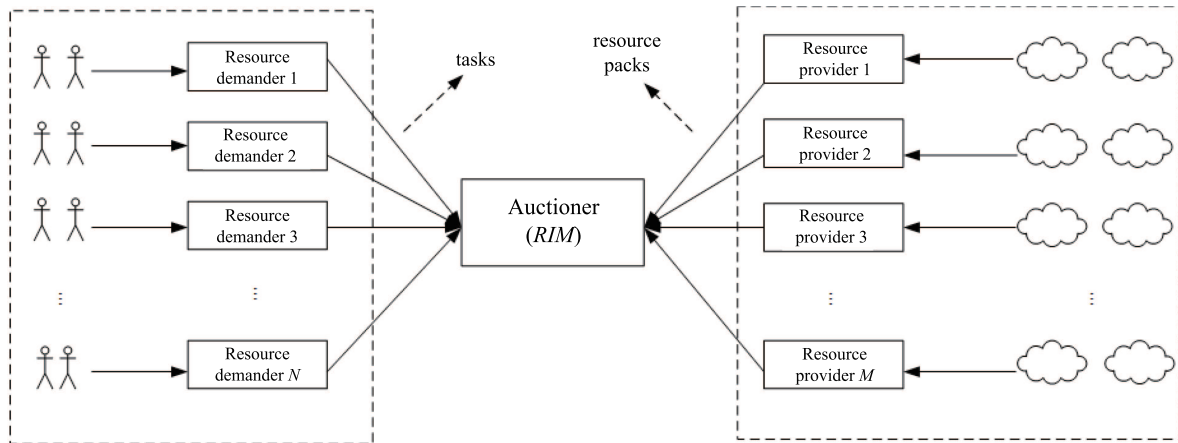


FIGURE 2. Auction relationship diagram

Assuming that in an auction time period T_n , as shown in Figure 2, N resource demanders apply to RIM for renting resources, and resource demander r_i sends a task package T_i to RIM , including n_i user tasks, and the user task collection is represented by $T_i = \{T_{i,1}, T_{i,2}, \dots, T_{i,n_i}\}$. Each user task is represented by a quadruple, that is, $T_{i,j} = \{budget_{i,j}, baseBid_{i,j}, taskComp_{i,j}, deadline_{i,j}\}$. $budget_{i,j}$ represents the unit computing resource budget of task $T_{i,j}$, and the task bid cannot exceed its budget. $baseBid_{i,j}$ represents the reserve price of unit computing resources of task $T_{i,j}$. $taskComp_{i,j}$ represents the

amount of computing resources that task $T_{i,j}$ needs to rent (in GHz). $deadline_{i,j}$ represents the deadline for task $T_{i,j}$. At the same time, M resource providers apply to RIM for willingness to rent resources, and resource provider p_k sends resource package S_k to RIM . The cloudlet resource package collection is represented by $S_k = \{S_{k,1}, S_{k,2}, \dots, S_{k,m_k}\}$. p_k has m_k cloudlets willing to rent resources. The cloudlet collection of p_k is represented by $C_k = \{c_{k,1}, c_{k,2}, \dots, c_{k,m_k}\}$, and each cloudlet resource package is represented by a two-tuple, that is, $S_{k,q} = \{cost_{k,q}, cloudletComp_{k,q}\}$. $cost_{k,q}$ represents the cost of computing resources per unit of $c_{k,q}$. $cloudletComp_{k,q}$ represents the maximum computing resource demand of cloudlet resources that $c_{k,q}$ can provide.

3.1. Problem description. In the auction process, the trading parties try to maximize their own benefits to conduct the auction, and the benefits of the resource provider p_k are expressed as the difference between the transaction price and the cost of the rental resources of its cloudlet collection, that is,

$$PU_k = \sum_{q=1}^{m_k} \sum_{i=1}^N \sum_{j=1}^{n_i} Rent_{i,j}^{k,q} \cdot taskComp_{i,j} \cdot \left(price_{i,j}^{k,q} - cost_{k,q} \right) \quad (1)$$

The benefits of the resource demander r_i are expressed as the difference between the bidding price and the transaction price of renting resources for its task collection, that is,

$$RU_i = \sum_{j=1}^{n_i} \sum_{k=1}^M \sum_{q=1}^{m_k} Rent_{i,j}^{k,q} \cdot taskComp_{i,j} \cdot \left(bid_{i,j} - price_{i,j}^{k,q} \right) \quad (2)$$

In order to optimize resource allocation and pricing, it can be transformed into linear integer programming to model and determine the resource pricing problem. The goal is to maximize the total benefits of the cloudlet federation. The optimization problem can be expressed as

$$\max \left(\sum_{i=1}^N RU_i + \sum_{k=1}^M PU_k \right) \quad (3)$$

s.t.

$$Rent_{i,j}^{k,q} \in \{0, 1\}, \forall 1 \leq i \leq N, \forall 1 \leq j \leq n_i, \forall 1 \leq k \leq M, \forall 1 \leq q \leq m_k \quad (4)$$

$$price_{i,j}^{k,q} \geq cost_{k,q}, \forall 1 \leq i \leq N, \forall 1 \leq j \leq n_i, \forall 1 \leq k \leq M, \forall 1 \leq q \leq m_k \quad (5)$$

$$price_{i,j} \leq bid_{i,j}, \forall 1 \leq i \leq N, \forall 1 \leq j \leq n_i \quad (6)$$

$$\sum_{i=1}^N \sum_{j=1}^{n_i} Rent_{i,j}^{k,q} \cdot taskComp_{i,j} \leq cloudletComp_{k,q}, \forall 1 \leq k \leq M, \forall 1 \leq q \leq m_k \quad (7)$$

$$\sum_{k=1}^M \sum_{q=1}^{m_k} Rent_{i,j}^{k,q} \leq 1, \forall 1 \leq i \leq N, \forall 1 \leq j \leq n_i \quad (8)$$

where Equation (4) indicates whether the task $T_{i,j}$ is leased to the cloudlet $c_{k,q}$. If $Rent_{i,j}^{k,q} = 1$, task $T_{i,j}$ is leased to the cloudlet $c_{k,q}$. Equation (5) ensures that the transaction price of unit computing resources $price_{i,j}^{k,q}$ is not lower than the cost of unit computing resources of the cloudlet $cost_{k,q}$. Equation (6) guarantees the transaction price of unit computing resources $price_{i,j}$ is not higher than the bid for the unit computing resource of the task $bid_{i,j}$. Equation (7) indicates that the amount of computing resources $\sum_{i=1}^N \sum_{j=1}^{n_i} Rent_{i,j}^{k,q} \cdot taskComp_{i,j}$ rented by the cloudlet $c_{k,q}$ must not be higher than the

total amount of computing resources $cloudletComp_{k,q}$ that $c_{k,q}$ can provide. Equation (8) ensures that a task $T_{i,j}$ can only be rented to one cloudlet $c_{k,q}$.

3.2. Pricing strategy. The resource provider's cloudlets should not only ensure the maximization of benefits, but also maintain the service quality of the system. Its asking price increases with the increase of its resource utilization rate. When the resource utilization rate of the cloudlet is low, the asking price is reduced to improve self-interest. When the resource utilization rate is high, increasing the asking price is to reduce the number of tasks received and the service pressure of cloudlets, but the asking price must be higher than its cost.

Assuming that at time t , RIM selects cloudlet $c_{k,q}$ that can provide services for task $T_{i,j}$, and the resource utilization rate of $c_{k,q}$ is $\frac{usedComp_{k,q}^t}{cloudletComp_{k,q}}$, then the asking price of the unit computing resources of $c_{k,q}$ is

$$ask_{k,q}^t = cost_{k,q} \cdot \left(1 + \left(\frac{usedComp_{k,q}^t}{cloudletComp_{k,q}} \right)^\beta \right) \quad (9)$$

where $usedComp_{k,q}^t$ indicates the amount of computing resources that $c_{k,q}$ has used at time t . $cloudletComp_{k,q}$ indicates the maximum amount of computing resources that $c_{k,q}$ can provide at the beginning of the auction, and β indicates the influence factor of resource utilization rate on the asking price, $0 \leq \beta \leq 1$.

Dibaj et al. [26] believed that the bid of the resource demander is related to the task deadline. The earlier the task deadline, the higher the task bid. Therefore, the bid of the unit computing resource of $T_{i,j}$ is

$$bid_{i,j} = baseBid_{i,j} + \frac{budget_{i,j} - baseBid_{i,j}}{deadline_{i,j} + 1} \quad (10)$$

If the amount of computing resources that cloudlet can rent meets the resource requirements of the task and the bid price of the task is not lower than the asking price of the cloudlet, then the auction of task $T_{i,j}$ and cloudlet $c_{k,q}$ can be successful, recording $Rent_{i,j}^{k,q} = 1$. After the auction, the transaction price of the unit computing resources is expressed as

$$price_{i,j} = price_{i,j}^{k,q} = \theta \cdot bid_{i,j} + (1 - \theta) \cdot ask_{k,q} \quad (11)$$

where θ represents the price adjustment factor of the transaction price, $0 \leq \theta \leq 1$.

If $T_{i,j}$ still cannot find a matching cloud after L rounds of auctions, then $T_{i,j}$ will trade with the remote cloud. The asking price per unit of computing volume of the remote cloud is $cloudPrice$. If the bid of the task is not lower than the asking price of the remote cloud, the transaction price of the unit computing resource is expressed as

$$price_{i,j} = \theta \cdot bid_{i,j} + (1 - \theta) \cdot cloudPrice \quad (12)$$

3.3. Price adjustment strategy. Assuming that the bid of task $T_{i,j}$ in the l th round of auction is $bid_{i,j}^l$, the bid will gradually increase with the increase of the number of auctions, but will not exceed the task budget $budget_{i,j}$. In order to protect the benefit of the resource demander, then in the $(l+1)$ -th round of auction, the bid of the unit computing resource of the task is expressed as

$$bid_{i,j}^{l+1} = bid_{i,j}^l + \delta (budget_{i,j} - bid_{i,j}^l) \quad (13)$$

where δ represents the price adjustment factor of the resource demander, $0 \leq \delta \leq 1$.

Assuming that the bid of cloudlet $c_{k,q}$ in the l -th round of auction is $ask_{k,q}^l$, and the asking price gradually decreases with the increase of the number of auctions, but will not

be lower than its cost, then in the $(l + 1)$ -th round of auction, the asking price of the unit computing resources of $c_{k,q}$ is expressed as

$$ask_{k,q}^{l+1} = ask_{k,q}^l - \gamma (ask_{k,q}^l - cost_{k,q}) \quad (14)$$

where γ represents the price adjustment factor of the resource provider, $0 \leq \gamma \leq 1$.

3.4. Resource pricing process based on double auction. Based on the analysis of the model, combined with the pricing adjustment strategies and transaction rules of resource demanders and resource providers, the resource pricing process based on double auction is as follows.

Step 1: Each of the N resource demanders receives n_i user tasks and organizes them into one task package, and then submits task information to *RIM*. At the same time, each of the M resource providers receives m_k cloudlet resource packages and organizes them into one resource package, submit the resource information to *RIM*.

Step 2: The auctioneer receives N task packages from resource demanders and M resource packages from resource providers. The auctioneer builds the *cloudletList* and *taskList* according to the time of task submission.

Step 3: The auctioneer calculates the bidding price of the task and the asking price of cloudlets according to the dynamic pricing strategy, and determines the winning task and cloudlet through the auction rules. If the task cannot find a matching cloudlet, it will enter the next round of auction by adjusting the price.

Step 4: Repeat Step 3. If after the L rounds of auctions, the tasks that have not rented resources choose to trade with the remote cloud. If all tasks in the transaction process can rent resources or the auction time has come, the auction will end. After the auction is over, the auction result is returned to the corresponding resource demander and resource provider.

3.5. Resource pricing process based on double auction. In the cloudlet federation, we propose a resource pricing algorithm (RPA) based on double auction. The algorithm has three components: 1) the pricing problem of tasks and cloudlets, and the appropriate cloudlet group is selected for each task; 2) the auction of tasks and cloudlets, and we select the optimal cloudlet for tasks according to the auction and price adjustment rules; 3) the auction of tasks and remote cloud. The algorithm is outlined as Algorithm 1. It can be seen that $T_{i,j}$ selects cloudlets from *cloudletList* and constructs $cl_{i,j}$ according to the principle of $userComp_{i,j} \leq cloudletComp_{k,q}^t$, and the time complexity of this is $O\left(\sum_{k=1}^M m_k\right)$. If $T_{i,j}$ has undergone L rounds of price adjustment, then the maximum time complexity for $T_{i,j}$ to find matching cloudlets is $O\left(L \cdot \sum_{k=1}^M m_k\right)$. There are $\sum_{i=1}^N n_i$ tasks in the cloudlet federation, so to complete the matching of tasks and cloudlets, the maximum time complexity is $O\left(L \cdot \sum_{i=1}^N n_i \cdot \sum_{k=1}^M m_k\right)$.

4. Experiment Analysis. Through computer simulation experiments, this paper compares the dynamic resource pricing algorithm (RPA) we designed with the uniform pricing algorithm (UPA) [16] and the fixed proportion pricing algorithm (FPPA) [17] to verify the effectiveness of the algorithm. The experimental platform is IntelliJ IDEA2019, and we set consistent experimental parameters according to these references. The experimental parameters are set as follows: the number of cloudlets per resource provider is 5, the total amount of computing resources of the cloudlet is [5, 20] GHz, the unit computing resource cost of the cloudlet is 5, the cloud computing resource capacity is 1000 GHz, and the asking price of computing resources per cloud is 10. The number of user tasks per

Algorithm 1. Resource pricing algorithm based on double auction

Input: $T_{i,j}$, $cloudletList$
Output: $price_{i,j}^{k,q}$, $Rent_{i,j}^{k,q}$

1. Initialize $cl_{i,j}$, $al_{i,j}$
 2. According to the principle of $userComp_{i,j} \leq cloudletComp_{k,q}^t$, select those cloudlets that can provide services for the task from $cloudletList$, and construct the winning cloudlet sequence $cl_{i,j}$
 3. Calculate the asking price of each cloudlet from $cl_{i,j}$ for the task $T_{i,j}$ using Equation (1)
 4. Construct the winning asking price list $al_{i,j}$, and then calculate the bid price of $T_{i,j}$ using Equation (2)
 5. **for** $l \leftarrow 1$ to L
 6. Eliminate the cloudlet whose asking price is higher than the bid of $T_{i,j}$ in $cl_{i,j}$
 7. Sort the remaining list in ascending order of the asking price
 8. **if** some cloudlets with the same asking price exist **then**
the cloudlet $c_{k,q}$ with the lowest cost will be selected
 9. Update the resource amount of the wining cloudlet resource package, that is,
 $cloudletComp_{k,q} = cloudletComp_{k,q} - taskComp_{i,j}$
 10. Calculate $price_{i,j}^{k,q}$ according to Equation (11) recording $Rent_{i,j}^{k,q} = 1$
 11. **if** $T_{i,j}$ has found a matching cloudlet **then**
break
 12. **else**
 $T_{i,j}$ and cloudlets will modify the price according to Equation (13) and Equation (14)
 13. **end for**
 14. return $price_{i,j}^{k,q}$, $Rent_{i,j}^{k,q}$
-

resource demander is 50, the amount of computing resources for user tasks is [0.2, 1] GHz, user task deadline is [50, 200] ms, the reserve price of computing resources per unit of user tasks is [5, 10], the influence factor of resource utilization rate on the asking price β is 0.5, the price adjustment factor of the resource demander δ is 0.5 and the price adjustment factor of the resource provider γ is 0.5.

Figure 3 describes the changes of the total benefits of the three algorithms under different numbers of resource demanders when the number of resource providers is 10. As can be seen from the figure, the total benefits of the cloudlet federation obtained by our proposed resource pricing algorithm (RPA), fixed proportion pricing algorithm (FPPA) and uniform pricing algorithm (UPA) first increase with the increase of the number of resource demanders. The maximum value is reached at 16, and then decreases as the number of resource demanders increases, and the RPA algorithm consistently achieves higher total benefits than the other two algorithms. The algorithm RPA can dynamically adjust the price according to the resource utilization rate and task deadline of the cloudlet, which improves the bidding competitiveness of cloudlets with low resource utilization rate and tasks with long deadlines, and increases the number of tasks that are successfully auctioned, so the total benefits of the cloudlet federation obtained by RPA are the highest. The number of resource demanders increases, and the number of tasks

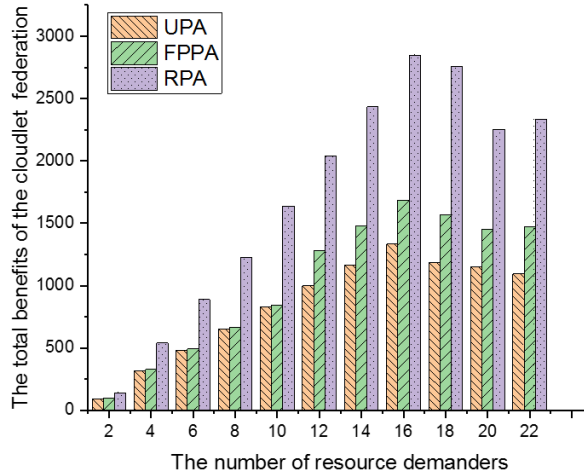


FIGURE 3. The influence of the number of resource demanders on the total benefits

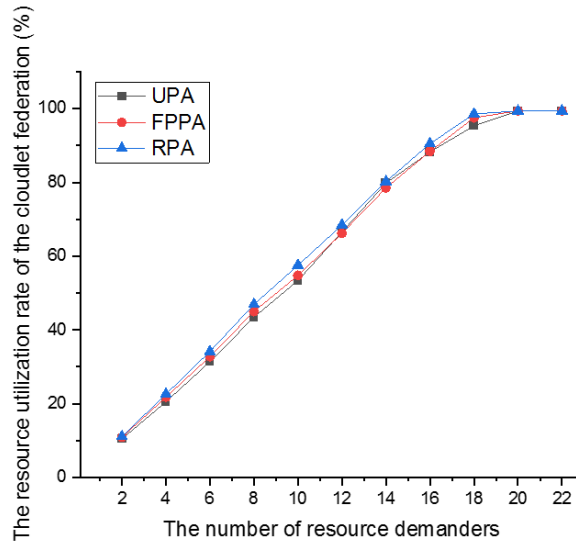


FIGURE 4. The influence of the number of resource demanders on resource utilization rate

requesting resources also increases, resulting in an increase in the total benefits. However, when the number of resource demanders reaches a certain number, the number of resource providers is limited, and tasks can only be auctioned for resources at a higher transaction price, which reduces their own benefits and thus reduces the total benefits. In summary, the experimental analysis shows that the algorithm RPA has higher performance in improving the total benefits of the cloudlet federation, and the best performance is when the number of resource demanders is 16.

Figure 4 describes the changes in the resource utilization rate of the cloudlet federation for the three algorithms under different numbers of resource demanders when the number of resource providers is 10. It can be seen from the figure that the resource utilization rate of the cloudlet federation obtained by the three algorithms increases with the increase of the number of resource demanders. As the number of resource demanders increases, the number of tasks requesting resources also increases, and more resources are leased out, thereby increasing resource utilization rate. When the number of resource demanders ranges from 2 to 22, the resource utilization rate of the RPA algorithm is always no lower than the other two pricing algorithms. Among them, when the number

of resource demanders is from 8 to 12, the resource utilization rate of the algorithm RPA is significantly improved compared with the other two algorithms. After the number of resource demanders exceeds 20, the resource utilization rate of the cloudlet federation of the three algorithms approaches 100%. In the algorithm RPA, the asking price of cloudlets takes account of the resource utilization rate, and adjusts the price according to the resource supply and demand in real time, while the other two algorithms do not consider this. Therefore, when the number of resource demanders is small, the algorithm RPA has higher resource utilization rate than the other two algorithms.

Figure 5 describes the changes of the total benefits of the cloudlet federation of the three algorithms under different numbers of resource providers when the number of resource demanders is 10. It can be seen from the figure that the total benefits of the algorithm RPA first increase with the increase of the number of resource providers, reach the highest value when the number of resource providers is 10, and then decrease with the increase of the number of resource providers. The total benefits of the RPA algorithm are always higher than that of the two pricing algorithms, and the variation of the RPA algorithm is also greater than that of the two algorithms. When the number of resource providers increases, the number of cloudlets that can provide resources increases, and more tasks rent resources, so that the total benefits of the cloudlet federation increase. When the number of resource providers exceeds a certain number, the number of resource demanders is limited, and the resource provider's cloudlets can auction tasks at a lower asking price, but it also reduces its own benefits, resulting in a decrease in the total benefits of the cloudlet federation. Since the tasks of the algorithms UPA and FPPA and cloudlets are quoted in a fixed way, there are fewer choices, and the auction is easy to stop, so the degree of change in the total benefits is relatively small. However, the algorithm RPA can dynamically adjust the bids of tasks according to task deadlines and price adjustment factors, while cloudlets adjust dynamically the asking prices according to resource utilization rate, and the total benefits of maintaining the auction continue to vary greatly. To sum up, the experiments show that the RPA algorithm has better performance in improving the total benefits of the cloudlet federation, and the best performance is when the number of resource providers is 10.

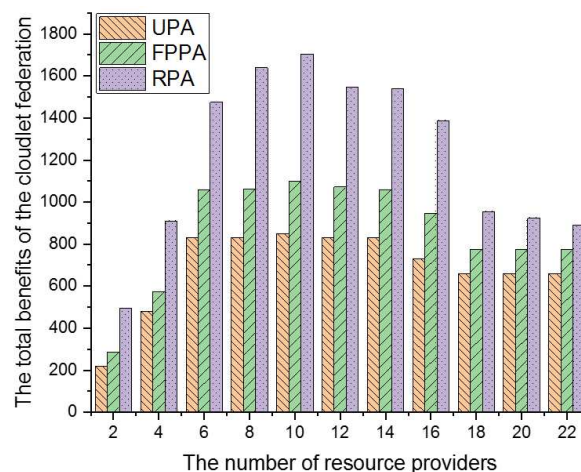


FIGURE 5. The influence of the number of resource providers on the total benefits

Figure 6 describes the changes in the ratio of tasks successfully auctioned for the three algorithms under different numbers of resource providers when the number of resource demanders is 10. As can be seen from the figure, when the number of resource providers is 2, 4, 6, and 8, the number of successful auction tasks of the three algorithms increases

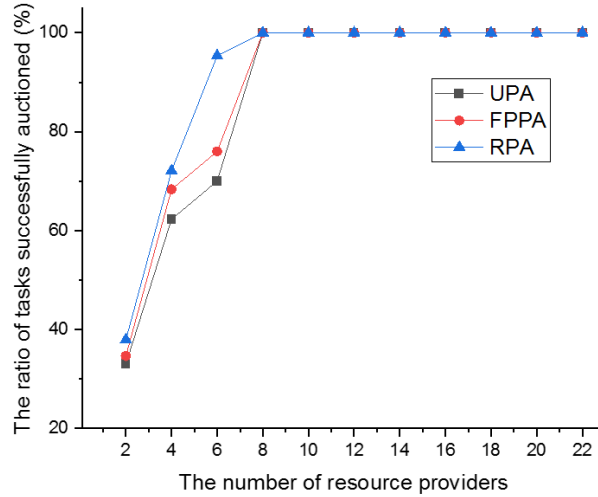


FIGURE 6. The influence of the number of resource providers on the ratio of the number of tasks successfully auctioned

as the number of resource providers increases, and the number of resource providers which can provide services increases. The number of cloudlets has also increased, and more tasks can be successfully auctioned. When the number of resource providers is 8, the proportion of tasks that can be successfully auctioned by the three algorithms is 100%, and all tasks can be successfully auctioned. After the number of resource providers exceeds 8, the ratio of the number of successful auction tasks for the three algorithms remains unchanged. When the number of resource providers is small, the task can choose from fewer cloudlets. The algorithms UPA and FPPA have fewer tasks for successful auctions due to the limitations of fixed bids. In the pricing mechanism of algorithm RPA, cloudlets considers resource utilization rate, and tasks consider task deadlines, which improves the flexibility of auctions, and can dynamically adjust prices in real time to win more auctions. As the number increases, the system is in a state of oversupply, and the selection of tasks expands. All three pricing algorithms can make the auction of all tasks successful. Therefore, when the number of resource providers is small, the algorithm RPA has better performance in terms of the number of tasks that are auctioned successfully.

Figure 7 shows the change of the total benefits of the algorithm RPA with five different transaction price impact factors θ under different numbers of resource demanders when the number of resource providers is 10. As can be seen from the figure, due to the increase in the number of resource demanders, the tasks that require resources also increase. Therefore, when the number of resource demanders ranges from 2 to 18, the total benefits of the five different transaction price influencing factors of the algorithm RPA of the cloudlet federation increase as the number of resource demanders increases. When the number of resource demanders is 18, the total benefits obtained by the algorithm RPA of the five transaction price influencing factors reach the highest value. After the number of resource demanders exceeds 20, the total benefits obtained by the algorithm RPA of the five different transaction price influencing factors are reduced, but the changes are relatively small. When $\theta = 0$, the total benefits of the cloudlet federation obtained by the RPA algorithm are always higher than the other four impact factors. The smaller θ is, the more the transaction price is more inclined to the asking price of the resource provider. The lower the transaction price that the resource demander needs to pay, the higher the benefits the resource demander can get, while the change in the benefits obtained by the resource provider is small. The total benefits increase with the increase of the benefits of

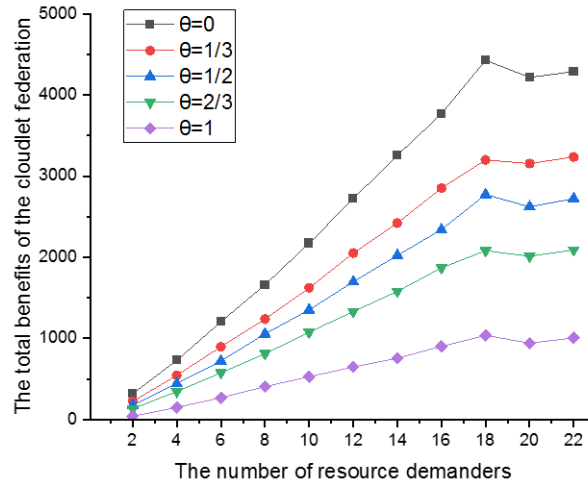


FIGURE 7. The influence of the number of resource demanders on the total benefits with different θ

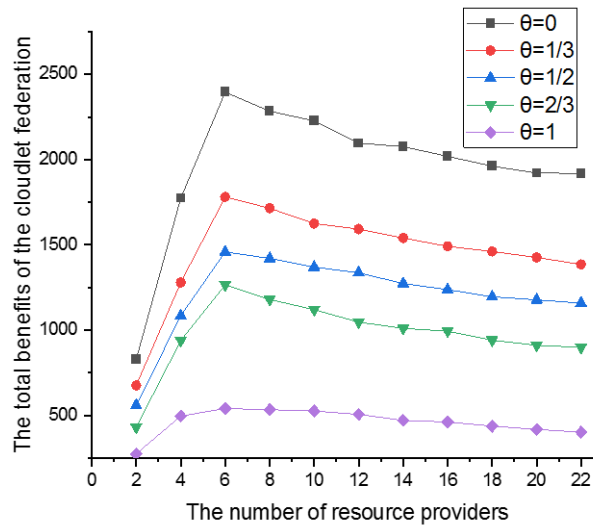


FIGURE 8. The influence of the number of resource providers on the total benefits with different θ

the resource demander. Therefore, when $\theta = 0$, the total benefits of the cloudlet federation obtained by algorithm RPA are the highest. However, as the number of resource demanders increases, the number of resource providers is limited, and the task can only be successfully auctioned at a higher transaction price. The benefits of resource demanders are reduced, and the degree of increase in the benefits of resource providers is less than that of resource demanders. Therefore, when the number of resource demanders reaches a certain number, the total benefits of the cloudlet federation will also decrease. A review of the experimental analysis shows that when the transaction price impact factor $\theta = 0$, the total benefits of the cloudlet federation are the highest.

Figure 8 shows the change of the total benefits of the algorithm RPA with five different transaction price impact factors θ under different numbers of resource providers when the number of resource demanders is 10. It can be seen from the figure that the total benefits of the algorithm RPA with five different transaction price influencing factors first increase with the increase of the number of resource providers. When the number of resource providers is 6, the total benefits reach the highest value. Then the total benefits

of the cloudlet federation decrease with the increase of the number of resource providers. As the number of resource providers increases, the state of the system is from excess demand to oversupply. The smaller θ is, the more the transaction price is more inclined to the asking price of the resource provider, and the higher the benefits for the resource demander will be. Therefore, when the system is in oversupply, the transaction price impact factor $\theta = 0$ is higher than the total benefits of the cloudlet federation obtained by the other four transaction price impact factors. However, when the number of resource providers exceeds a certain number, the total benefits of the cloudlet federation decrease as the number of resource providers increases. This is because the number of resource providers has increased, the choice of resource demanders has expanded, and the cloudlets of resource providers need to ask for lower resource prices to be successful in the auction. The increase of the benefits of the resource demander is smaller than the decrease of the benefits of the resource demander, which leads to the decrease of the total benefits of the cloudlet federation. A review of the experimental analysis shows that when the transaction price impact factor $\theta = 0$, the total benefits of the cloudlet federation are the highest.

5. Conclusion. To solve the problem of resource pricing when the cloudlet federation is renting, the paper, from the overall perspective of the cloudlet federation, considers resource utilization rate and task deadline, designs dynamic pricing and price adjustment strategies. In this paper, aiming at maximizing the total benefits of the cloudlet federation, a resource pricing model based on double auction is constructed, and the algorithm RPA is proposed to further verify the total benefits of RPA, UPA and FPPA. During the auction process, both parties can dynamically adjust the price and conduct multiple rounds of auctions to obtain higher total benefits of the cloudlet federation. The simulation results show that the algorithm RPA is significantly better than the fixed algorithms UPA and FPPA in terms of total benefits, resource utilization rate and auction task success rate in the cloudlet federation when the number of resource demanders or providers is small. In the algorithm RPA, when the transaction price influence factor $\theta = 0$, the total benefits obtained by the cloudlet federation are higher than other transaction influence factors. In the next research, trust mechanism would be introduced into the cloudlet federation to further improve the resource pricing mechanism.

Acknowledgment. This research was supported by the National Natural Science Foundation of China [grant number 62262011], and the Guangxi Key Laboratory Director Fund of Embedded Technology and Intelligent Information Processing [grant number 2020-1-7].

REFERENCES

- [1] K. Gai, M. Qiu, H. Zhao, L. Tao and Z. Zong, Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing, *Journal of Network and Computer Applications*, vol.59, pp.46-54, 2016.
- [2] M. Satyanarayanan, P. Bahl, R. Cáceres and N. Davies, The case for VM-based cloudlets in mobile computing, *IEEE Pervasive Computing*, vol.8, no.1, pp.14-23, 2009.
- [3] X. Cao, G. Tang, D. Guo, Y. Li and W. Zhang, Edge federation: Towards an integrated service provisioning model, *IEEE-ACM Trans. Networking*, vol.28, no.3, pp.1116-1129, 2020.
- [4] C. F. Jiang, L. C. Duan and C. L. Liu, VRAA: Virtualized resource auction and allocation based on incentive and penalty, *Cluster Comput.*, vol.16, pp.639-650, 2013.
- [5] W. Wang and X. Ling, An approach to allocate grid resources based on auction, *Computer Applications*, vol.29, no.11, pp.25-28, 2010.
- [6] B. Tan, S. Xu, G. Xu and Y. Zhou, Optimal parking space allocation based on combinatorial auction and uniform price, *Journal of Transportation Systems Engineering and Information Technology*, vol.21, no.3, pp.193-199, 2021.

- [7] J. He, S. Tao, Y. Deng, L. Chen and Z. Mou, Research on multi-sensor resource dynamic allocation auction algorithm, *Journal of Northwestern Polytechnical University*, vol.37, no.2, pp.330-336, 2019.
- [8] C. Sun, T. Zheng, L. Chen, Y. Xie, B. Gao and S. Mei, Energy storage sharing mechanism based on combinatorial double auction, *Power System Technology*, vol.44, no.5, pp.1732-1739, 2020.
- [9] J. Wang, X. Chen and Y. Wang, A secure sealed-bid electronic auction scheme, *Acta Electronica Sinica*, vol.31, no.10, pp.1578-1579, 2003.
- [10] Q. Chen, Y. Shi, X. Yang and L. Tang, Resource scheduling mechanism for virtual network slice based on online double auction, *Journal of Electronics & Information Technology*, vol.40, no.7, pp.1738-1744, 2018.
- [11] H. Baghban, C. Y. Huang and C. H. Hsu, Latency minimization model towards high efficiency edge-IoT service provisioning in horizontal edge federation, *Multimedia Tools and Applications*, vol.81, no.9, pp.26803-26820, 2021.
- [12] S. Moon and Y. Lim, Task migration with partitioning for load balancing in collaborative edge computing, *Applied Sciences-Basel*, vol.12, no.3, 1168, DOI: 10.3390/app12031168, 2022.
- [13] T. Zhang, A QoS-aware virtual resource pricing service based on game theory in federated clouds, *Int. J. Intelligent Systems Technologies and Applications*, vol.19, no.2, pp.191-206, 2020.
- [14] B. Pourebrahimi, K. Bertels, G. M. Kandru and S. Vassiliadis, Market-based resource allocation in grids, *The 2nd IEEE International Conference on e-Science and Grid Computing*, pp.80-88, 2006.
- [15] F. Huang, H. Ye and W. Hao, Cost-aware resource management based on market pricing mechanisms in edge federation environments, *The Journal of Supercomputing*, 2022.
- [16] Y. Wu, T. Cai and D. Zhang, Computing power trading and pricing in mobile edge computing based on Stackelberg game, *Journal of Computer Applications*, vol.40, no.9, pp.2683-2690, 2020.
- [17] H. Zhang, F. Guo, H. Ji and C. Zhu, Combinational auction based service provider selection in mobile edge computing networks, *IEEE Access*, vol.5, pp.13455-13464, 2017.
- [18] Y. Luo and S. Qiu, Optimal resource reservation scheme for maximizing profit of service providers in edge computing federation, *2019 IEEE Intl. Conf. on Dependable, Autonomic and Secure Computing*, pp.876-879, 2019.
- [19] U. Awada and J. Zhang, Edge federation: A dependency-aware multi-task dispatching and co-location in federated edge container-instances, *2020 IEEE International Conference on Edge Computing (EDGE)*, Beijing, China, pp.91-98, 2020.
- [20] Y. Lin, D. Truong, A. Ali, C. Li, Y. Lai and T. T. Dinh, Proxy-based federated authentication: A transparent third-party solution for cloud-edge federation, *IEEE Network*, vol.34, no.6, pp.220-227, 2020.
- [21] D. Kelaidonis, A. Rouskas, V. Stavroulaki and P. Demestichas, A federated edge cloud-IoT architecture, *2016 European Conference on Networks and Communications (EuCNC)*, Athens, Greece, pp.230-234, 2016.
- [22] N. M. Randriamasinoro, K. K. Nguyen and M. Cheriet, Optimized resource allocation in edge-cloud environment, *2018 Annual IEEE International Systems Conference (SysCon)*, pp.1-8, 2018.
- [23] F. Teng and F. Magoulès, Resource pricing and equilibrium allocation policy in cloud computing, *2010 10th IEEE International Conference on Computer and Information Technology (CIT2010)*, pp.195-202, 2010.
- [24] Z. Hu and Y. Liu, Dynamic resource pricing based on combinatorial double auction in cloud environment, *Computer Engineering*, vol.38, no.8, pp.19-21, 2012.
- [25] D. Sun, G. Chang, C. Wang, Y. Xiong and X. Wang, Efficient Nash equilibrium based cloud resource allocation by using a continuous double auction, *2010 International Conference on Computer Design and Applications*, pp.94-99, 2010.
- [26] S. M. R. Dibaj, A. Miri and S. Mostafavi, A cloud priority-based dynamic online double auction mechanism (PB-DODAM), *Journal of Cloud Computing – Advances, Systems and Applications*, vol.9, no.1, pp.1-26, 2020.
- [27] S. K. Yadav and R. Kumar, A scalable and utility driven profit maximized auction of resources model for cloudlet based mobile edge computing, *Wireless Personal Communications*, vol.119, no.1, pp.527-565, 2021.
- [28] M. Mihailescu and Y. M. Teo, Dynamic resource pricing on federated clouds, *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp.513-517, 2010.
- [29] H. Robertus, C. L. Hendiarta and A. A. S. Gunawan, A systematic literature review: Internet of Things and cloud-based application for medical sector, *ICIC Express Letters, Part B: Applications*, vol.12, no.8, pp.669-677, 2021.

- [30] J. Xu and B. Palanisamy, Cost-aware resource management for federated clouds using resource sharing contracts, *2017 IEEE 10th International Conference on Cloud Computing*, pp.238-245, 2017.
- [31] Q. Wang, Y. Mao, Y. Wang and L. Wang, Computing task offloading based on multi-cloudlet collaboration, *Journal of Computer Applications*, vol.40, no.2, pp.328-334, 2020.
- [32] Z. Li, *Research on Resource Allocation Algorithms Based on Double Auction in Mobile Edge Computing*, Graduation Thesis, Jiangxi Normal University, 2017.
- [33] K. Cheng, Y. Teng, W. Sun, A. Liu and X. Wang, Energy-efficient joint offloading and wireless resource allocation strategy in multi-MEC server systems, *2018 IEEE International Conference on Communications (ICC)*, pp.1-6, 2018.

Author Biography



Hengzhou Ye received the B.S. and Ph.D. degrees from Guilin University of Technology and Guangxi University, Guangxi, China, in 2002 and 2019, respectively. He is currently a Full Professor with Guilin University of Technology. His research interests include age of information, service composition, mobile edge computing, and multi-objective optimization.



Yihui Chen received the B.S. degree from Zhaoqing College, Guangdong, China, in 2019. She is currently pursuing the M.S. degree at Guilin University of Technology, Guangxi, China. Her research interests include age of information and mobile edge computing.



Xinxiao Li received the B.S. and M.S. degree from Guangxi University and Guilin University of Technology Guangxi, China, in 2001 and 2007, respectively. She is currently a lecturer in Guilin University of Technology. Her research interests include service composition, mobile edge computing and affective computing.