Pricing the Cloud: A QoS-based Optimal Design

Yang Lu[1], Jiacheng Dong[1,*]

Abstract

Cloud computing has a major impact on the IT industry. How to price and allocate cloud resources to meet users' requirements is an important problem. This paper proposes a dynamic mechanism to pricing cloud services, which can work in complex environments such as distributed system and uncertain budget constraints. A direct relationship between QoS and price is established. The approach uses an optimization technique to estimate the potential transaction price in the distributed network. It can allocate cloud resources under uncertainties, where providers can optimize their revenues, and consumers can obtain the resources at a relatively low price.

Keywords: QoS (quality of service), Reliability, Availability, Robust optimization auction, Optimal revenue, Budget constraint

Article History:
Received November 05, 2025
Revised January 20, 2026
Accepted March 20, 2026
Available Online March 30, 2026

I. Introduction

"Cloud computing: a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." Cloud services include Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS) [Armbrust, et al., 2010]. Cloud service can be treated as an ordinary commodity that has spread through the Internet. Currently, leading companies are gearing up to use cloud services for their businesses: Amazon's AWS (Amazon Web Services), Google's GAE (Google App Engine), Microsoft's Azure, and IBM's Cloud. For instance, Amazon Elastic Compute Cloud (Amazon EC2) offers seven instance purchasing options: On-demand Instances, Reserved Instances, Scheduled Instances, Spot Instances, Dedicated Hosts, Dedicated Instances, and Capacity Reservations. But they have not yet provided roust optimal design directly related to QoS metrics (reliability and availability). Based on the SLA (service level agreement), the overall performance of cloud service is guaranteed to some extent, but the relationship between price and QoS metrics still isn't clear. Usually, customers can purchase cloud service via various strategies, at certain prices which are determined by costs and by an economy proxy (supply and demand). As a special commodity, the cloud not only has costs that are similar to those of other goods, but it also has QoS-related costs that aren't easy to estimate. It is possible that customers bid on cloud services based on the QoS metrics that both provider and customer really care about.

Cloud computing provides us with on-demand and remote QoS-embedded services that are scalable, elastic, complex, and potentially decentralized [Zhang, et al., 2010]. Quality of service is crucial to the cloud industry. On the one hand, providers seek to improve the overall performance of cloud service to compete in the industry; on the other hand, customers expect cloud service to have high QoS standards, especially in its reliability and availability. The reliability is actually the probability that a system will be operational in a given time interval without any failures [Armbrust, et al., 2010]. The availability is actually the probability that the system will be up and function correctly in a certain time period [Armbrust, et al., 2010]. High availability, with high costs and price, is essential to guarantee QoS, to maintain customer's confidence, and to attract more customers. More customers are involved in the use of cloud services, at a lower price each customer needs to pay. In other words, it is the economies of scale. The percentile of multiple Nines and Fives is defined to express the different level of availability in general, e.g., 3 Nines is 99.90% and 4N5 is 99.995%. In our study, we will briefly illustrate the relationship between reliability and availability by MTBF (Mean Time between Failure) and MTTR (Mean Time to Repair). A balance between availability and price then will be established as the reference for participants to estimate the price of cloud services.

One customer's requirement for a specific service is not independent of those of other customers. Indeed, a customer using a specific service has some connection with the needs of other customers. In an auction, cloud services are auctioned concurrently, and bidders can compete for services. Since QoS is important, both for the provider and the customer, there is the potential to add QoS metrics (availability) into the pricing algorithms. A mechanism allocates services, associated with the expected QoS, to customers.

Depending on the customers' different requirements for cloud services, suppliers need a viable and efficient pricing mechanism that is critical to the allocation and optimization of the available cloud resources. For suppliers, customers will bid for their own budget for better service, and the suppliers will

[1]School of Computing and Artificial Intelligence, Beijing Technology and Business University, Beijing, China 102401

[*]Email: dreamjc213@gmail.com

https://doi.org/10.63646/VBHS9335

select customers based on their bids, provide resources for higher-priced customers and guarantee the cloud's QoS. For customers, they can freely choose resources according to their own needs. However, due to the liquidity of marginal customers, whose bidding is uncertain, and it is difficult to estimate the demand based on existing bids. The cloud resource scheme based on uncertain bid auction has the potential to improve both economic efficiency and optimal profits.

Our design is a QoS-based robust optimization mechanism that includes multiple customers bidding for a variety of cloud services with QoS guarantees offered by different providers in a distributed network. The reasons why our pricing model combines QoS metrics with auction design are (1) cloud computing is a new and advanced computing technology. Cloud services are different from ordinary products or services, which can be evaluated relatively easily based on supply and demand. However, the cloud has unique QoS features, such as reliability and availability. An auction is a proper approach to evaluate not-easy-estimated parameters of cloud. (2) Cloud pricing is still in its early development stage (Zheng, et al., 2014). There is no direct reference to pricing cloud services based on QoS metrics. Hence, an auction algorithm is an appropriate way to estimate the valuation of certain cloud services and to allocate resources efficiently. (3) An auction design can sell services in a way that customers come to expect, specifically based upon Service Level Agreement (SLA) and Quality of Service (QoS). There are no obvious QoS-based pricing algorithms in the extant literature. The most likely exponential relationship expresses the intrinsic connection between price and QoS metrics and offers a straightforward pricing reference to both provider and customer.

In this paper, we address the issue of designing a robust optimal mechanism, through which cloud services are distributed between provider and customer, along with QoS. The goals are (1) to propose QoS-embedded availability to price and to allocate cloud services from providers to customers, and (2) to flexibly set up a dynamic pricing mechanism to enable budget constraint. Our contribution is to build a QoS-based dynamic auction model to effectively allocate cloud resources. Specifically, as a critical indicator of QoS, availability is used to estimate the value of cloud service in the pricing algorithms. Because of the unique features of the cloud, the relationship between price and QoS availability is not simply linear. We apply a robust optimal scheme to allocate resources and to optimize revenues among participants.

The paper has the following structure. A complete overview of the auction design for cloud computing (fixed pricing strategy, dynamic pricing strategy, and auction pricing strategy) and discussion of QoS-embedded optimal pricing strategy are addressed in Section II. Section III depicts the relationship between QoS metrics (availability) and price. We describe and explain the detailed stages of functions and scheme in Section IV. Section V explores future directions. The final section summarizes the paper.

II. CLOUD PRICING MECHANISM

Cloud-related usage is becoming more and more popular in the daily life. Pricing is one of the important issues in the cloud industry. In this paper, we divided cloud pricing mechanisms into three categories: fixed pricing strategies, dynamic pricing strategies, and auction pricing strategies. Specifically, the three popular fixed pricing strategies are pay-as-you-go, subscription, and pay-for-use. Dynamic pricing strategies have many different algorithms, such as Genetic Model, Financial Optional Theory, Markov Decision Process, etc. Auction pricing strategies consist of various auction design, such as English auction, sealed-bid auction, double auction, and combinatorial auction. A summary of the cloud pricing scheme is addressed in Table 1.

A. Fixed Pricing Strategy

Provider presents the price of cloud service, and customer pays the amount if the customer expects to use the service. The price is stable. Although the fixed pricing strategy is straightforward to both provider and customer, the mechanism is unfair to all potential customers who don't have the same needs [Yeo, et al., 2010]. Even to the provider, the mechanism is not an optimal strategy, especially when demand is higher than supply.

Pay-as-you-go, which charges customers based on their overall resource usage. The customers pay for the amount they consume of a product or the amount of time they use a certain service. Pay for resources, which charge customers based on the technical features of cloud services, such as storage and bandwidth. Subscription, customers subscribe to certain service for a time period. Subscription is another type of fixed pricing, in which the customer pays a fixed amount of money to use the service for longer periods at any convenient time or amount.

The provider presents the price of cloud service, and if a customer wishes to use the service, the customer pays the amount according to the agreement. In the cloud industry, three pricing mechanisms are widely employed, pay-as-you-go, subscription, and pay-for-use. Pay-as-you-go charges customers based on the overall resource usage for a certain time period. Subscription is another type of fixed pricing, in which a customer pays a fixed amount in advance to use the service for a longer period of time. Depending on the technical features of cloud services, such as storage and bandwidth, a customer is required to make pay fees. The common feature of these three pricing mechanisms is that price remains stable over the contact time. While fixed pricing strategies are straightforward to both provider and customer, the mechanism is inappropriate to all customers because not all customers have the same needs [Al-Roomi, et al., 2013; Luong, et al., 2017]. Even for providers, the mechanism is not optimal, especially when demand is higher than supply in the real world. Our pricing design is embedded with an auction algorithm that can offer all customers equal opportunities and competition associated with dynamically changing prices.

The most prevalent method of pricing in cloud is pay per use, which is based on Units with constant price. Another common pricing model is subscription, whereby users sign a contract (subscribe) based on the constant price of service unit and a

longer period of time, for example six months or a year. Obviously, customers and providers would like to use static and simple pricing models in order to ease payment prediction. Nevertheless, for high-value services dynamic pricing can be more efficient.

B. Dynamic Pricing Strategy

The final price is calculated by a dynamic pricing mechanism. In this pricing strategy, prices change with respect to market conditions or status. An appropriate pricing algorithm is an effective way to estimate the value of cloud service and to allocate the available resources. Aiming at providing high QoS, Sharma, et al. (2012) employed the financial option theory and the related economic model to capture the real value of cloud service within the lower and upper boundaries.

Macías & Guitart (2011) proposed a genetic model for pricing in cloud computing markets. Choosing a good pricing model via their genetic algorithms involved three main steps: define a chromosome, evaluate it, and finally select the best pairs of chromosomes for reproduction and discarding those with the worst results. The results of the simulation illustrated that genetic pricing acquired the highest revenues in most of the scenarios. Service providers employing genetic pricing achieved revenues up to 100% greater than the other dynamic pricing strategies and up to 1000% greater than the fixed pricing strategy. The proposed genetic model with a flexible genome was proven to be more stable against noise and earned more money than the one with the rigid genome. The proposed genetic model is easy to implement, flexible, and easily adapted to a set of various parameters that influence pricing. The genetic pricing approach can be further explored by defining relations between the parameters that influence pricing.

Li, et al. (2011) proposed a pricing algorithm for cloud computing resources. This proposal used the cloud bank agent model as a resource agency because it could provide proper analysis and assistance for all members. The authors used a price update iterative algorithm to determine the price. It analyzed the historical utilization ratio of the resources, iterated current prices constantly, assessed the availability of resources for the next round, and determined the final price. The model included a user request broker (GCA), cloud banking, a cloud service agent (CSA), and a cloud resource agent (GRA). The proposed pricing model was comparatively fixed because it could not adapt to the rapid changes that typically occur in the market. However, it could reduce the costs to providers and maximize their revenues, allowing resources to be used more effectively.

In particular, George, et al., (2019) study practices of AWS spot instances—spot prices are changed dynamically by AWS based on real-time demand and idle capacity. Truong-Huu & Tham (2014) consider competition between service providers and formulate the competition through a Markov Decision Process. However, those studies have not examined and compared the pricing schemes that are considered in our study, and they have not studied the service providers' optimal choices of pricing schemes and the impact of the chosen pricing schemes on the customers' choices of service providers.

Another major recent research on cloud pricing is dynamic pricing algorithms. The market-clearing price is computed by a dynamic pricing mechanism. In this pricing strategy, prices change with respect to market conditions or status. An appropriate pricing algorithm is an effective way to estimate the value of cloud service and to distribute the available resources. The dynamic pricing mechanism can accurately estimate the price based on market status and present a reasonable price for both provider and customer. For instance, Macías & Guitart (2011) proposed a Genetic Model for pricing in cloud computing markets. Sharma, et al. (2012) employed the Financial Option Theory and the related economic model to capture the real value of cloud service within the lower and upper boundaries. Truong-Huu & Tham (2014) used a Markov Decision Process to illustrate the competition between providers and to compute the price. Dynamic pricing strategy mainly focuses on changing prices based on supply and demand, but the strategy ignores the important cloud feature, especially the QoS metrics. The value of clouds is directly affected by QoS metrics and operational process. Although dynamic pricing strategy can offer a reasonable price, it is difficult for practitioners to estimate the actual price of cloud, because there are no QoS-related parameters in the extant dynamic pricing designs. Our pricing design added QoS availability into pricing models that clearly indicates how a customer can bid directly on the expected QoS level with the related price.

C. Auction Pricing Strategy

Theoretically, prices should be determined by the interaction between supply and demand; however, in practice, it seems like the companies employ specific auction mechanisms in setting up the prices, in order to make optional profits and to attract more potential customers. From Amazon's initial effort of using auction-based allocation, it is reasonable to expect that cloud providers will be interested in more efficient allocation and pricing schemes in the near future.

This is a market mechanism for service allocation that enables users and providers to deal through double-sided combinational auction. This mechanism is suitable for cases requiring various services and where many participants exist. Both users and service providers should be satisfied by the resource allocation mechanism. A double-sided auction model and K-pricing scheme were used in this mechanism.

Basically, all participants accurately know the nature of the available resources and the distribution of these possible resources. Each participant, whether the participant is a provider or a customer, needs a bidding strategy [Milgrom & Weber, 1982]. Reasonable uncertainties from participants and markets make auction design more attractive and more desirable through some mathematical algorithms and bidding strategies.

Many classic auction designs were employed to price cloud services. When they were applied to pricing cloud services, the mechanism would be NP-hard. Wang, et al., (2012) proposed an English auction. The mechanism focuses on how to maximize the seller's revenue and shorten the execution time. Another study about dynamic auctions was conducted by Wang,

et al., (2013) to cope with changes in cloud market. Asymptotic optimization, incentive compatibility, and computational complexity were pursued, through their capacity allocation scheme. A double auction model has been used to distribute cloud resources to multiple buyers and sellers [Shang, et al., 2010]. A framework uses marginal bid to evaluate cloud capacity distribution and to generate reasonable revenue for the cloud provider [Lin, et al., 2010].

In practice, leading cloud service companies, such as Amazon AWS or Microsoft Azure, lack the adoption of a desirable auction mechanism to efficiently allocate resources. For instance, Amazon EC2 offers three categories of pricing schemes to sell its cloud services: Reserved Instances, On-Demand Instances (Fixed Price), and Spot Instances. Spot Instance is one type of auction-style pricing policy, through which bidders can periodically bid on offered resources dynamically. Successful bidders can use these instances until the auction price exceeds their bids at a later date. This design has attracted significant attention from practitioners and researchers, and it has prompted a number of studies on auction-based pricing design. Zaman & Grosu (2013) believed that an auction design is better than the current fixed-price mechanism. Depending on the experimental results, Auction-Greedy is a better choice than Auction-Linear Programming. Auction-Greedy can generate higher revenue and resource utilization in a limited time. But these mechanisms are only applied to certain types of items. Extended from Zaman & Grosu (2013), Samimi, et al., (2016) proposed the Combinatorial Double Auction Resource Allocation, focusing on the economic efficiency of the cloud computing environments. Providers are more concerned about what economic profits they can obtain. Thus, based on individual rationality and incentive compatibility, our study will address two other important issues, QoS-based pricing mechanisms and revenue approximation.

Table I
Summary of Pricing Scheme for Cloud Allocation

Fixed Pricing Strategy
Classification: Pay-as-you-go, Subscription, Pay for Resource
Characteristics: Stable price for the time of usage. Easily implemented, without an auctioneer.
Example: Amazon EC2, Microsoft Azure
Dynamic Pricing Strategy [Macías & Guitart, 2011; Sharma, et al., 2012; Du, et al., 2013; Truong-Huu & Tham, 2014]
Classification: Genetic Model, Financial Option Model, Markov Decision Process
Characteristics: Price is changed based on supply and demand. Multiple providers to multiple customers.
Auction Pricing Strategy
English Auction [Wang, et al., 2012]
Characteristics: One-sided ascending auction. The winning buyer pays the bidding price. One single provider to multiple customers, with an auctioneer.
Double Auction [Zaman & Grosu, 2013; Samimi, et al., 2016]
Characteristics: Both provider and customer propose bids. Multiple providers to multiple customers, with an auctioneer
Combinatorial Auction [Zaman & Grosu, 2013; Samimi, et al., 2016]
Characteristics: Customer can bid for customized bundle cloud services. The auction design construct computation complexity. Multiple providers to multiple customers, with an auctioneer

In a cloud environment, practitioners are concerned about Quality of Service (QoS). The sharing of resources, software, and information makes up the basic functions of cloud computing. In an auction design, economic models can be used to reduce unnecessary costs, to regulate available resources, to provide incentives for providers, and to stimulate buyers to choose the preferred services that are associated with rational evaluation and QoS criteria. Due to the heterogeneity of cloud computing, a well-designed auction is a basic step toward allocating available resources to meet supply and demand, via the Internet. QoS-based auction design is a good attempt at finding an efficient and effective way to benefit from cloud computing and to guarantee the performance of the cloud.

Quality of Service describes the requirements that a service provider should provide to its customers, such as service availability, reliability, security, privacy, scalability, and integrity. Our model allows bidders to submit their own valuations and QoS-based bidding strategies. The transaction price will be calculated based on the basis of each participant's bid directly related to QoS availability, although the economic perspectives of supply and demand in the cloud market still impact participants' auctioning strategies. What we did was use an exponential function to price cloud services based on QoS metrics (availability) and bids from customers. In this way, available cloud services can be distributed to clients with heterogeneous QoS expectations and portfolio requirements. Only buyers submit bids and QoS preferences for a certain item. A QoS-based auction mechanism is a good pricing strategy that could enrich the classic auction design, especially for the bidding targets with unique or not-easy-to-estimate features, e.g., cloud service has QoS criteria.

III. THE MODEL OF QOS AND PRICE

A. Reliability and Availability

The reliability and availability of cloud computing are the crucial QoS parameters, but they are difficult to quantifiably analyze because cloud services are implemented through the entire serving process in a complicated network that integrates software, hardware, and the related techniques. Reliability focuses on the downtime of cloud service, while availability represents the uptime of cloud services [Armbrust, et al., 2010]. Reliability and availability are interrelated with each other. A better performance of reliability will lead to better performance of availability. A more stable system will decrease the number of failures and lessen the time of repair. In the real world, reliability and availability are integrated with each other greatly impacting the overall performance of cloud services.

The relationship between reliability and availability is:

$$A = \frac{MTBF}{MTBF+MTTR} = \frac{Uptime}{Uptime+Downtime} \quad (1)$$

A refers to availability. MTBF (Uptime) and MTTR (Downtime) are the two parameters of reliability. MTBF is the Mean Time between Failure, and MTTR is the Mean Time to Repair. Thus, based on the above function, the availability can be increased either by increasing the average time interval between repairs (MTBF) or by decreasing the repair time (MTTR). The intuitive way to represent availability is using the downtime associated with Nines and Fives. The following table

is an example.

Table II
Representation of Availability

Availability	Downtime
99% (2-nines)	3.65 days/year
99.5% (2N5)	1.825 days/year
99.9% (3-nines)	8.76 hours/year
99.95% (3N5)	4.38 hours/year
99.99% (4-nines)	52 minutes/year
99.995% (4N5)	26 minutes/year
99.999% (5-nines)	5 minutes/year

B. The Relationship between Availability and Price

The QoS metrics is the availability of cloud service ($A_{(x)}$), and the bidding price is the average price (y) of the same item from different customers' bids.

The function that expresses availability is:

$$A_{(x)} = 1 - 10^{-x+g(x)} \quad (2)$$

Where x represents the level of availability as Nines and Fives, e.g., x=2, it means "2Nines" as 99%; x=3.5, it means "3N5" as 99.95%. x = {x/x=2, 2.5, 3, 3.5, 4, 4.5, …}. $g(x)$ is a function that is the potential to calculate the result of x. Since we know all possible (x, y): (2, 99%), (2.5, 99.5%), (3, 99.9%), …, the function can be addressed as:

$$A_{(x)} = 1 - 10^{-x+(0.5+\ln 0.5)sin^2(\pi x)} \quad (3)$$

Therefore, x is expressed by $A_{(x)}$ as:

$$x = \begin{cases} -\ln[1 - A_{(x)}], if\ x\ is\ an\ integer \\ (0.5 + \ln 0.5) - \ln[1 - A_{(x)}], if\ x\ is\ not\ an\ integer \end{cases}$$

C. One Example of The Relationship between Availability and Price

The exponential function that expresses price by the QoS availability is:

$$y = a + cb^{h_{(x)}} \quad (4)$$

Where, a is the constant and c is the slope. b is the base of the exponentiation, and $h_{(x)}$ is the exponent. Now, we can compute the exponential function based on x and y. In fact, there exist many different ways to establish the relationship between QoS metrics and cloud price. This relationship can be represented by a linear function that is easy to implement but not accurate. Because even with a slight increase in reliability and availability, the costs associated with cloud services increase significantly. Hence, an exponential algorithm is a better approach to describe the relationship between QoS metrics and cloud price. Figure 1 illustrates one example of the exponential relationship between price and availability. Based on the different levels of availability (x) that customers bid, such as 2 (99%), 2.5 (99.5%), 3 (99.9%), 3.5 (99.95%), 4 (99.99%), 4.5 (99.995%), 5 (99.999%), 5.5 (99.9995%), 6 (99.9999%), firstly the average price (y) of each level of cloud service is calculated accordingly: \$0.004, \$0.01, \$0.0146, \$0.0194, \$0.027, \$0.0342, \$0.0676, \$0.0937, \$0.1415. Then, based on function (4), the relationship between QoS availability and price is calculated as:

$$y = 0.001e^{0.8203x}$$

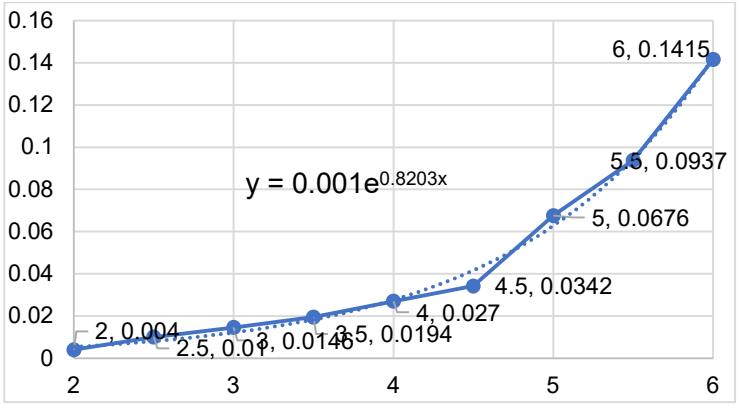

Figure I One Example of the Exponential Relation

IV. ROBUST OPTIMIZATION SCHEME

A. Robust Auction Design

Symbols and Assumptions

This section addresses the related parameters and mathematical symbols used. Suppose there are n customers, n = {*1, 2, …, n*}, and *ith* represents the customer *i*. There are m providers, m = {*1, 2, …, m*}, and *jth* represents the resource *j*. $T_{m,j}$ represents the *mth* supplier offers the resource *j* or not. $T_{m,j}$ is a binary factor, $T_{m,j}$ (=1) means that the *mth* supplier is capable of offering the resource *j*, $T_{m,j}$ (=0) means that the *mth* supplier doesn't provide the resource *j*. $w_{i,j,m}$ illustrates that the customer *i* obtains the resource *j* provided by supplier *m*, and all the bidding resources are set in the matrix *W*.

In the system, the customer is the buyer, the provider is the seller. The buyer's bid is $b_{i,j,m}$, represented by vector B, and the final price is $p^b_{i,j,m}$. The customer *i*'s value is V_i, $v_{i,j,m}$ represents the valuation of the resource *j* provided by the seller *m*. The cost of the resource *j* is $c_{i,j}$. $D_{i,j}$ represents the demand of the buyer *i* on the resource *j*. The capability of the resource *j* from the seller *m* is $C_{j.m}$. The more the buyer bids, the utility (μ) is higher. Specifically, $\mu_{i,j} = h_{i,j}v_i\ (b_{i,j}) - c_i\ (b_{i,j})$.

The customer's bid has two general constraints. (1) a customer needs to bid incrementally, (2) all customers are truthful bidders. The set (U) of all possible bids is: U_j = {$(b_{1,j}$, $b_{2,j}$, …, $b_{n,j}$) | -Γ' ≤ $\frac{\sum_{i=1}^{n} v_{i,j} - n\mu_j}{\sqrt{n}\,\sigma_j}$ ≤ Γ'}. Where μ_j and σ_j are the bidding prices of two different times, respectively.

Optimal Auction Model

The overall auction model (*M1*) is:

$$max \sum_i \sum_j \sum_m x_{i,j,m} b_{i,j,m} \quad (5)$$

Subject to

$$\sum_m x_{i,j,m} = 1, \forall\, i,j \quad (6)$$

$$\sum_i x_{i,j,m} D_i T_{m,j} \le C_{j,m}, \forall\, m,j \quad (7)$$

$$\sum_m \sum_j \overline{b_{i,j}} - \overline{p_i} \le \sum_m \sum_j b_{i,j} x_{i,j,m} - p_i \quad (8)$$

$$c_{i,j,m} \le b_{i,j,m} \le v_{i,j,m}, \forall\, i,j,m \quad (9)$$

$$x_{i,j,m} \in \{0,1\}, \forall\, i,j,m \quad (10)$$

Function M1 is the maximum revenue for all the bidding resources. Constraint (6) indicates that there must be at least one seller (m) can provide the resource j as the buyer (i) expects. Constraint (7) indicates that the possible allocation of a seller' s resources is less than the seller's capacity. Constraint (8) indicates that all buyers are trustworthiness. Constraint (9) indicates that the bid is between the cost and the value of the resource.

Function *M1* is an integer (0-1) programming problem. This problem belongs to NP-hardness [Karp, 1972]. It is transformed to a solvable linear programming problem to find the optimal solution.

B. Solution to the Model

Algorithm 1

The *M1* model is based on the fact that $b_{i,j,m}$ can be estimated and controlled, but in reality, each user's pricing strategy and actual bid are dynamic and difficult to estimate. This study focuses on how to allocate resources to users based on their bidding strategies to meet the needs of users and resource providers. For the problem of uncertain optimization, a robust optimization method can be used to solve the problem. There are four criteria: optimal coordination based on probability, optimistic criteria, conservative criteria, and optimistic and conservative weighting [Beyer & Sendhoff, 2007; Bertsimas, et al., 2011; Bandi & Bertsimas, 2014]. This study uses conservative criteria. It is seeking the worst result of the original problem and then finding the optimal case under the worst scenario.

Since $b_{i,j}$ is uncertain in model *M1*, $\widehat{b_{i,j,m}}$ is used to represent the conservative bid of user *i*, and the user *i* has to pay this amount to get the required resource *j*. Constraint (8) can be converted to its equivalent. In order to achieve a reasonable resource allocation, each customer first proposes a conservative bid, and then provider can obtain a higher return based on the bid. Using a conservative bid ($\widehat{b_{i,j,m}}$) converts the original 0/1 problem into a linear programming problem. *M1* is converted to *M2*:

$$max \sum_i \sum_j \sum_m x_{i,j,m} \widehat{b_{i,j,m}} \tag{11}$$

Subject to

$$\begin{gathered} \sum_m x_{i,j,m} = 1, \forall\, i,j \\ \sum_i x_{i,j,m} v_{m,j} \le B_{i,m}, \forall\, m,j \\ \sum_i x_{i,j,m} D_i T_{m,j} \le C_{j,m}, \forall\, m,j \end{gathered} \tag{12}$$

$$\sum_m x_{i,j,m} \widehat{b_{i,j,m}} \le \sum_m \sum_j x_{i,j,m} \mu_{i,j,m}, \forall\, i,j \tag{13}$$

$$c_{i,j,m} \le \widehat{b_{i,j,m}} \le v_{i,j,m}, \forall\, i,j,m \tag{14}$$

$$x_{i,j,m} \ge 0, \forall\, i,j,m \tag{15}$$

Now, *M2* is converted into *M3*, which is the dual problem of *M2*.

$$\begin{aligned} min \sum_i \sum_j \varepsilon_{i,j} + \sum_i \sum_m (\delta_{m,j} B_{i,m} \\ + \theta_{m,j} \sum_j x^*_{i,j,m} U_{i,j}) \end{aligned} \tag{16}$$

Subject to

$$\varepsilon_{i,j} + z_{i,j,m}\delta_{m,j} + z_{i,j,m}\theta_{m,j} \ge z_{i,j,m} \tag{17}$$

$$\varepsilon, \delta, \theta \ge 0 \tag{18}$$

$z_{i,j,m}$ is the bid of the worst scenario, $x^*_{i,j,m}$ is the solution. The user i's uncertain bid is $U_{i,j}$. Specifically,

$$U_i = argmin \sum_m \sum_j x^*_{i,j,m} \mu_{i,j} \tag{19}$$

The user's conservative bid ($\widehat{b_{i,j,m}}$) can be represented by $z_{i,j,m}$. Since $z_{i,j,m}$ is the bid under the worst scenario,

$$\widehat{b_{i,j,m}} = \varepsilon^*_{i,j} + z_{i,j,m}\delta^*_{m,j} + z_{i,j,m}\theta^*_{m,j} \tag{20}$$

ε^*, δ^*, and θ^* are the parameters of the dual problem, respectively. These parameters can be solved by model M3. After the results of M3 are solved, the algorithm 1 can be employed to solve $\widehat{b_{i,j,m}}$.

Algorithm 1. Estimated Bidding Algorithm

Algorithm 1 Calculating the Estimated Bidding Price

Input: The Uncertain Price Set U

Output: $\widehat{b_{i,j,m}}$ and $x^*_{i,j,m}$

Based on model M2, calculate the bid of the worst scenario ($z_{i,j,m}$) and the associated allocation result ($x_{i,j,m}$);
Based on model M3, calculating the parameters ε^*, δ^*, and θ^*;
To each of the involved suppliers
Based on Function (20), calculate the estimated bidding price ($\widehat{b_{i,j,m}}$).

Algorithm 2

After we get $\widehat{b_{i,j,m}}$ and $x^*_{i,j,m}$, the Algorithm 2 will solve the allocation and the final price. The intermediary variable $y^b_{i,j,m}$ represents adopted allocation, and the final allocation result is $a^b_{i,j,m} = x^*_{i,j,m} + y^b_{i,j,m}$, $a^b_{i,j,m}$ is the final allocation results. Specifically, bidding vector is B, the final price is $p^b_{i,j,m}$. $y^{b-k}_{i,j,m}$ is the temporary intermediate allocation.

$$y^b_{i,j,m} = argmax \sum_i \sum_j \sum_m y_{i,j,m}(b - \widehat{b_{i,j,m}}) \tag{21}$$

$$y^{b-h}_{i,j,m} = argmax \sum_{i \ne k} \sum_j \sum_m y_{i,j,m}(b - \widehat{b_{i,j,m}}) \tag{22}$$

The user's flexible bidding policy:

$$P = arg \left\{ \begin{matrix} \sum_i \sum_m y_{i,j,m} \le 1 - \sum_i \sum_m x^*_{i,j,m} \\ \sum_j y_{i,j,m} \mu_{i,j,m} \le B_{i,m} - \sum_j x^*_{i,j,m} \widehat{b_{i,j,m}} + \sum_j x^*_{k,j,m} \theta^* U_{i,j,m} \end{matrix} \right\} \tag{23}$$

Since the final price is determined by the user *i*'s bid and other users' bids, we define Q_k is the allocation of the user *k*.

$$Q_k = arg \left\{ \begin{matrix} \sum_{i \ne k} \sum_m y_{i,j,m} \le 1 - \sum_i \sum_m x^*_{i,j,m} \\ \sum_j y_{i,j,m} \mu_{i,j,m} \le B_{i,m} - \sum_j x^*_{i,j,m} \widehat{b_{i,j,m}} \end{matrix} \right\} \tag{24}$$

Function (21)'s constraint is function (23), and function (22)'s constraint is function (24).

The final allocation ($a^b_{i,j,m}$) is

$$a_{i,j,m}^{b} = x_{i,j,m}^{*} + y_{i,j,m}^{b} \quad (25)$$

The final price (($p_{k,m}^{b}$) is

$$p_{k,m}^{b} = \sum_j y_{k,j,m}^{b}\widehat{b_{\iota,j,m}} + \sum_j x_{k,j,m}^{*}\widehat{b_{k,j,m}} - \sum_j x_{k,j,m}^{*}\theta^{*}U_{k,j,m} + \sum_{i\neq k}\sum_j\sum_m y_{i,j,m}^{b-k}(b - \widehat{b_{\iota,j,m}}) - \sum_{i\neq k}\sum_j\sum_m y_{i,j,m}^{b}(b - \widehat{b_{\iota,j,m}}) \quad (26)$$

Algorithm 2. Resource Allocation Algorithm

Algorithm 2 Resource Allocation Algorithm

Input: The bidding vector b and the estimated prices $\widehat{b_{\iota,j,m}}$ and $x_{i,j,m}^{*}$

Output: The final allocation vector $a_{i,j,m}^{b}$ and the final price $p_{i,j,m}^{b}$

To each supplier m

If $b_{i,j,m} \notin U_{i,j}$

Won't provide any resource to the user I;

Based on function (21) and (22), calculate the allocation $y_{i,j,m}^{b}$ and the temporary variable $y_{i,j,m}^{b-k}$ that satisfies function (23) and (24);

Based on Function (25) and (26), calculate the final allocation vector $a_{i,j,m}^{b}$ and the final price $p_{i,j,m}^{b}$.

The solution to M2 is solved by Algorithm 1 and Algorithm 2. We get the final allocation $a_{i,j,m}^{b}$ and the final price $p_{i,j,m}^{b}$.

C. Theoretical Proof

In our study, the final price is similar to the transaction price in VCG [Vickrey, 1961; Varian & Harris, 2014]. The transaction price is associated with each users' bids, which consist of the winning bids and unsuccessful bids. In this section, we will illustrate that the solutions to Algorithm 1 and Algorithm 2 are also the solution for the original model *M1*.

Lemma 1. The unknown variables (z, x^{*}, ε^{*}, δ^{*}, θ^{*}, $\hat{b}$) of Algorithm 1 and Algorithm 2 satisfy the following relations.

$$\sum_m\sum_i x_{i,j,m}^{*} \le 1, \forall\, j \quad (27)$$

$$\sum_j x_{i,j,m}^{*} z_{i,j,m} \le B_{i,m}, \forall\, i,m \quad (28)$$

$$\sum_m\sum_j x_{i,j,m}^{*} z_{i,j,m} \le \sum_m\sum_j x_{i,j,m}^{*} z_{i,j,m}, \forall\, i \quad (29)$$

$$x_{i,j,m}^{*}\widehat{b_{\iota,j,m}} = x_{i,j,m}^{*} z_{i,j,m}, \forall\, i,j,m \quad (30)$$

$$\sum_j x_{i,j,m}^{*}\widehat{b_{\iota,j,m}} \le B_{i,m}, \forall\, i,m \quad (31)$$

$$\sum_i\sum_j\sum_m x_{i,j,m}^{*}\widehat{b_{\iota,j,m}} = \sum_i\sum_j \varepsilon_{i,j}^{*} + \sum_i\sum_m (\delta_{m,j}^{*} B_{i,m} + \theta_{m,j}^{*}\sum_j x_{i,j,m}^{*} U_{i,j}) \quad (32)$$

According to Functions (27)-(29), $z_{i,j,m}$ is a fixed term. The optimal model *M2* can be converted to the following:

$$max \sum_i\sum_j\sum_m x_{i,j,m} z_{i,j,m}$$

s.t.

$$\sum_m\sum_i x_{i,j,m} \le 1, \forall\, i,j$$
$$\sum_j x_{i,j,m} z_{i,j,m} \le B_{i,m}, \forall\, i,m$$
$$\sum_m\sum_j x_{i,j,m} z_{i,j,m} \le \sum_m\sum_j x_{i,j,m}\mu_{i,j,m}, \forall\, i \quad (33)$$

Since $x_{i,j,m}^{*}$ is one solution to the optimal problem, Function (33) can be transformed into another question of *M3*.

$$max \sum_i\sum_j\sum_m x_{i,j,m} z_{i,j,m}$$

s.t.

$$\sum_m\sum_i x_{i,j,m} \le 1, \forall\, i,j$$
$$\sum_j x_{i,j,m} z_{i,j,m} \le B_{i,m}, \forall\, i,m$$
$$\sum_m\sum_j x_{i,j,m} z_{i,j,m} \le \sum_m\sum_j x_{i,j,m}^{*}\mu_{i,j,m}, \forall\, i \quad (34)$$

If $\bar{x}$ is the solution of function (34), and x^{*} is the solution of function (33), x^{*} is also one solution of function (34).

$$\sum_m\sum_i\sum_j \bar{x}_{i,j,m} z_{i,j,m} \ge \sum_m\sum_j\sum_i x^{*}{}_{i,j,m} z_{i,j,m} \quad (35)$$

And

$$\sum_m\sum_j \bar{x}_{i,j,m} z_{i,j,m} \le \sum_m\sum_j x^{*}{}_{i,j,m}\mu_{i,j,m}, \forall\, i \quad (36)$$

Based on function (35) and (36),

$$\sum_m\sum_j\sum_i x^{*}{}_{i,j,m} z_{i,j,m} \le \sum_m\sum_j x^{*}{}_{i,j,m}\mu_{i,j,m} \quad (37)$$

Thus, $x^{*}{}_{i,j,m}$ is the optimal solution to Function (21). Function (34) can be converted to the following:

$$\sum_m\sum_j x^{*}{}_{i,j,m} z_{i,j,m} \le \sum_m\sum_j x^{*}{}_{i,j,m}\mu_{i,j,m} \quad (38)$$

Hence, $x^{*}{}_{i,j,m}$ is an optimal solution to Function (33). U_i is computed from Function (19). We have the following:

$$max \sum_i\sum_j\sum_m x_{i,j,m} z_{i,j,m}$$

s.t.

$$\sum_m\sum_i x_{i,j,m} \le 1, \forall\, i,j$$
$$\sum_j x_{i,j,m} z_{i,j,m} \le B_{i,m}, \forall\, i,m$$
$$\sum_m\sum_j x_{i,j,m} z_{i,j,m} \le \sum_m\sum_j x_{i,j,m}^{*} U_i, \forall\, i \quad (39)$$

Obviously, $x^{*}{}_{i,j,m}$ is the optimal solution to Function (39). The Dual Problem of Function (39) can be the following:

$$min \sum_i\sum_j \varepsilon_{i,j} + \sum_i\sum_m (\delta_{m,j} B_{i,m} + \theta_{m,j}\sum_j x_{i,j,m}^{*} U_{i,j}) \quad (40)$$

s.t.

$$\varepsilon_{i,j} + z_{i,j,m}\delta_{m,j} + z_{i,j,m}\theta_{m,j} \geq z_{i,j,m}, \varepsilon, \delta, \theta \geq 0 \quad (41)$$

ε, η, and θ represent the parameters of three constraints. Define ε^*, δ^*, θ^* are the values corresponding to the three parameters, when $x^*_{i,j,m}$ is the optimal solution. According to Complementary Relaxation Theorem, we have the following:

$$\begin{aligned} \varepsilon^* \sum_m \sum_i x_{i,j,m} &= \varepsilon^*, \forall\, j \\ \delta^* \textstyle\sum_j x_{i,j,m} z_{i,j,m} &= \delta^* B_{i,m}, \forall\, i, m \\ \theta^* \textstyle\sum_m \sum_j x_{i,j,m} z_{i,j,m} &\leq \theta^* \sum_m \sum_j x^*_{i,j,m} U_{i,j}, \forall\, i \\ x^*_{i,j,m}\left(\varepsilon_{i,j} + z_{i,j,m}\delta_{m,j} + z_{i,j,m}\theta_{m,j}\right) &= x^*_{i,j,m} z_{i,j,m} \end{aligned} \quad (42)$$

$x^*_{i,j,m}$ is the optimal solution to Function (34), according to the Strong Dual Theorem, we get,

$$\begin{aligned} \sum_i \sum_j \sum_m x^*_{i,j,m} z_{i,j,m} &= \sum_i \sum_j \delta^* \\ &+ \sum_i \sum_m (\delta^* B_{i,m} \\ &+ \theta^* \sum_j x^*_{i,j,m} U_{i,j}) \end{aligned} \quad (43)$$

Based on function (20), (42), and (43), Function (18) and (19) can be proofed. And **Lemma 1** can be proved as well.

V. FUTURE DIRECTION

Based on conventional auction design, we added QoS metrics (availability) and established a robust optimal auction to price and allocate cloud resources. For future research, there are two potential directions. An auction model could be converted to a combinatorial double auction design. In this way, customers will have the opportunity to bid for bundled (package) cloud services. The major cloud trading platform is centralized and controlled by companies. A blockchain-based decentralized P2P cloud trading platform is potential to play an important role in allocating cloud resources between customers without unnecessary third-party intervention. The decentralized P2P cloud trading system is a good complementary of the conventional trading system.

Providers will compete with each other and submit bids related to guaranteed QoS. The mathematical issue (NP-Hardness) and the computational complexity should be carefully considered as well [Roughgarden, 2010]. Another direction is to adjust the QoS metrics. We only employed the availability as the indicator of QoS in this study. Multiple indicators can be added to represent the exponential relationship between price and QoS. Such as security. More QoS metrics are added into the auction algorithm, more accurate and practical the estimates will be. The relationship between price and multiple QoS indicators will be worth investigating in real-world industries. Whereas more QoS metrics involved may lead to the complexity of algorithms, deep learning method is a possible way to assess cloud price, according to the training layer (historical pricing records) and output layers (estimated results).

VI. CONCLUSION

In the cloud environment, QoS criteria is critical to the overall performance of cloud resources. The extant research seldom illustrates the relationship between QoS metrics and price; our design has the potential to achieve this. By a dynamic pricing mechanism, cloud resources can be traded between provider and customer with uncertainties, e.g., budget constraints. In our work, we address the robust optimal allocation of cloud services to users who are flexible in their bids for their preferred QoS metrics (availability). Our study presents us with one probable way of investigating the questions of how companies formulate prices for cloud services and how customers can utilize the pricing mechanism to bid for cloud services, based upon the QoS metrics (availability). We highlighted the QoS indicator that was employed in the auction mechanism to price cloud services among different participants. The robust optimal auction design is an appropriate pricing mechanism for cloud resources in a distributed trading system.

REFERENCES

Al-Roomi, M., Al-Ebrahim, S., Buqrais, S., & Ahmad, I. (2013). Cloud computing pricing models: a survey. International Journal of Grid and Distributed Computing, 6(5), 93-106. DOI: 10.14257/ijgdc.2013.6.5.09.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50-58. DOI: 10.1145/1721654.1721672.

Bandi, C., & Bertsimas, D. (2014). Optimal design for multi-item auctions: A robust optimization approach. Mathematics of Operations Research, 39(4), 1012-1038. DOI: 10.1287/moor.2014.0645.

Bertsimas, D., Brown, D. B., & Caramanis, C. (2011). Theory and applications of robust optimization. SIAM review, 53(3), 464-501. DOI: 10.1137/080734510.

Beyer, H. G., & Sendhoff, B. (2007). Robust optimization–a comprehensive survey. Computer methods in applied mechanics and engineering, 196(33-34), 3190-3218. DOI: 10.1016/j.cma.2007.03.003.

Du, A. Y., Das, S., & Ramesh, R. (2013). Efficient risk hedging by dynamic forward pricing: A study in cloud computing. INFORMS Journal on Computing, 25(4), 625-642. DOI: 10.1287/ijoc.1120.0526.

George, G., Wolski, R., Krintz, C., & Brevik, J. (2019, June). Analyzing AWS spot instance pricing. In 2019 IEEE International Conference on Cloud Engineering (IC2E) (pp. 222-228). IEEE. DOI: 10.1109/IC2E.2019.00036.

Karp, R. (2021). Reducibility among combinatorial problems (1972). DOI: 10.7551/mitpress/12274.003.0038.

Li, H., Liu, J., & Tang, G. (2011, May). A pricing algorithm for cloud computing resources. In 2011 international conference on network computing and information security (Vol. 1, pp. 69-73). IEEE. DOI: 10.1109/NCIS.2011.22.

Lin, W. Y., Lin, G. Y., & Wei, H. Y. (2010, May). Dynamic auction mechanism for cloud resource allocation. In 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (pp. 591-592). IEEE. DOI: 10.1109/CCGRID.2010.92.

Luong, N. C., Wang, P., Niyato, D., Wen, Y., & Han, Z. (2017). Resource management in cloud networking using economic analysis and pricing models: A survey. IEEE Communications Surveys & Tutorials, 19(2), 954-1001. DOI: 10.1109/COMST.2017.2647981.

Macías, M., & Guitart, J. (2011, March). A genetic model for pricing in cloud computing markets. In Proceedings of the 2011 ACM Symposium on Applied Computing (pp. 113-118). DOI: 10.1145/1982185.1982216.

Milgrom, P. R., & Weber, R. J. (1982). A theory of auctions and competitive bidding. Econometrica: Journal of the Econometric Society, 1089-1122. DOI: 10.2307/1911865.

Roughgarden, T. (2010). Algorithmic game theory. Communications of the ACM, 53(7), 78-86. DOI: 10.1145/1785414.1785439.

Samimi, P., Teimouri, Y., & Mukhtar, M. (2016). A combinatorial double auction resource allocation model in cloud computing. Information Sciences, 357, 201-216. DOI: 10.1016/j.ins.2014.02.008.

Shang, S., Jiang, J., Wu, Y., Huang, Z., Yang, G., & Zheng, W. (2010). DABGPM: A double auction Bayesian game-based pricing model in cloud market. In Network and Parallel Computing: IFIP International Conference, NPC 2010, Zhengzhou, China, September 13-15, 2010. Proceedings (pp. 155-164). Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-15672-4_14.

Sharma, B., Thulasiram, R. K., Thulasiraman, P., Garg, S. K., & Buyya, R. (2012, May). Pricing cloud compute commodities: A novel financial economic model. In 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012) (pp. 451-457). IEEE. DOI: 10.1109/CCGrid.2012.126.

Truong-Huu, T., & Tham, C. K. (2014). A novel model for competition and cooperation among cloud providers. IEEE Transactions on Cloud Computing, 2(3), 251-265. DOI: 10.1109/TCC.2014.2322355.

Varian, H. R., & Harris, C. (2014). The VCG auction in theory and practice. American Economic Review, 104(5), 442-445. DOI: 10.1257/aer.104.5.442.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. The Journal of finance, 16(1), 8-37. DOI: 10.2307/2977633.

Wang, W., Liang, B., & Li, B. (2013, June). Revenue maximization with dynamic auctions in IaaS cloud markets. In 2013 IEEE/ACM 21st International Symposium on Quality of Service (IWQoS) (pp. 1-6). IEEE. DOI: 10.1109/IWQoS.2013.6550265.

Wang, X. W., Wang, X. Y., & Huang, M. (2012, May). A resource allocation method based on the limited english combinatorial auction under cloud computing environment. In 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (pp. 905-909). IEEE. DOI: 10.1109/FSKD.2012.6234231.

Yeo, C. S., Venugopal, S., Chu, X., & Buyya, R. (2010). Autonomic metered pricing for a utility computing service. Future Generation Computer Systems, 26(8), 1368-1380. DOI: 10.1016/j.future.2009.05.024.

Zaman, S., & Grosu, D. (2013). Combinatorial auction-based allocation of virtual machine instances in clouds. Journal of parallel and distributed computing, 73(4), 495-508. DOI: 10.1016/j.jpdc.2012.12.006.

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications, 1, 7-18. DOI: 10.1007/s13174-010-0007-6.

Zheng, X., Martin, P., Brohman, K., & Da Xu, L. (2014). CLOUDQUAL: a quality model for cloud services. IEEE transactions on industrial informatics, 10(2), 1527-1536. DOI: 10.1109/TII.2014.2306329.

APPENDIX

Table III. Main Variables and Explanations (Alphabetic Order)

Variable	Description	Variable	Description
a	the constant of the exponential function	$p^b_{i,j,m}$	the final price
$a^b_{i,j,m}$	the final allocation results	Q_k	the allocation of the user k
$A_{(x)}$	the QoS availability of cloud service	$v_{i,j,m}$	the valuation of the resource j provided by the seller m
b	the base of the exponential function	V_i	the valuation of customer i
$b_{i,j,m}$	the buyer's bid	$T_{m,j}$	the mth supplier offers the resource j or not
$\widehat{b_{i,j,m}}$	the conservative bid of user i	U	the set of all possible bids
c	the coefficient of the exponential function	$U_{i,j}$	user i's uncertain bid
$c_{i,j}$	The cost of resource j	x	the level of QoS availability
$C_{j.m}$	The capability of the resource j from the seller m	$x^*_{i,j,m}$	the decision variable, $x_{i,j,m} \in \{0,1\}$
$D_{i,j}$	the demand of the buyer i on the resource j	y	the average bidding price of cloud service
h(x)	the exponent of the exponential function	$y^b_{i,j,m}$	the intermediary variable
i	the ith bidder	$y^{b-k}_{i,j,m}$	the temporary intermediate allocation
j	the jth provider	$z_{i,j,m}$	the bid of the worst scenario
m	there are m sellers	μ	the utility
n	there are n buyers	$\varepsilon^*, \delta^*, \theta^*$	the parameters of the dual problem