A Utility Game Driven QoS Optimization for Cloud Services

Yan Wang, Jian-Tao Zhou, and Xiaoyu Song

Abstract—Cloud services request lower cost compared to traditional software of self-purchased infrastructure due to the characteristics of on-demand resource provisioning and pay-as-you-go mode. Current enterprises compact their business software as services into cloud platform to users. In the cloud services market, service providers attempt to make more profits from their services, while users hope to choose low-cost services with high-quality. The conflict of interests between users and service providers is an important challenge for the booming cloud service market. This article characterizes this application problem formally based on a utility game model of service providers and users. In the model, QoS is considered as the basis for determining the utilities of both parties from an economic point of view. By analyzing the behaviors of users and service providers, we introduce the concept of reputation cost for the first time in the model and find a QoS solution that balances the utilities of users and service providers in service transactions. In such a balance, any change in either party's strategy will result in a loss of utility. And then a QoS optimization method is designed to obtain a near-optimal QoS solution for a tradeoff between user satisfaction and provider profit. Extensive simulation experiments are conducted to substantiate the effectiveness of our method. The results are applicable to win-win service applications between service providers and users.

Index Terms—Cloud services, game, user preference, utility, QoS optimization

1 Introduction

Cloud computing can be regarded as a mode of resource use. Its main idea is to provide various resources to users in the form of services through virtualization technology. In terms of service level, it can be divided into IaaS, PaaS and SaaS. SaaS (Software-as-a-Service) is a mode of providing software through the Internet. A service provider uniformly deploys the application software to the cloud platform and provides services for users by renting the infrastructure resources of the cloud platform. A user can obtain various software services in the cloud according to their own needs without purchasing expensive software and hardware facilities. Moreover, software upgrade and maintenance can be completed by cloud service providers, which can greatly save users' costs for using software. With the popularization of cloud computing technology and the increase of service providers in the cloud platform, especially the increase of service providers providing similar software, it is an inevitable trend for end users to choose a service that can meet their functional requirements with the best QoS (Quality of Service) but the lowest price. For service providers, in order to be in a favorable position in the competition, they will improve QoS by adjusting their service resource scheduling scheme, so as to obtain higher user satisfaction. Obviously, the more resources are invoked, the more quality of service is provided. But it also means that service providers have to pay higher cost and get lower return at the same price. From this perspective, the interests of service providers and users are in conflict. The essence of this conflict lies in the contradiction between the cost of service and the quality of service. Under the constraint of service price, users aim to pursue services with high-quality, while service providers aim to provide services with the lowest cost. The conflict of interests between users and service providers is an important challenge for the booming cloud service market. How to find a balance between the interests of users and service providers is a problem to be solved in this paper.

Existing publications have done a lot of research work in improving the quality of service or reducing the cost of service, which to some extent improves users' satisfaction. Most of these works focused on one-sided interest of users or service providers, but the essence of service process is a kind of economic activity with interest constraint relationship between users and service providers. Therefore, when seeking a balance of interest conflict, the interests of both sides should be considered. This paper regards users and service providers as economic entities with limited rationality and independence in the cloud service market, establishes a game model between users and service providers, and seeks some strategies to balance the interests of both parties by solving the equilibrium solution of the game.

In the game, QoS is the focus of both sides. On the one hand, QoS determines the cost of service, and thus has an impact on the revenues of service providers. On the other hand, QoS also determines user satisfaction. Therefore,

- *Yan Wang and Jian-Tao Zhou are with the Ecological Big Data Engineering Research Center of the Ministry of Education, National and Local Joint Engineering Research Center of Mongolian Intelligent Information Processing, College of Computer Science, Inner Mongolia University, Hohhot 010021, China. E-mail: cswy@imu.edu.cn, cszhoujiantao@qq.com.*
- *Xiaoyu Song is with the Department of Electrical and Computer Engineering, Portland State University, Portland, OR 97207 USA. E-mail: songx@pdx.edu.*

Manuscript received 23 August 2019; revised 11 October 2020; accepted 22 February 2021. Date of publication 26 February 2021; date of current version 7 October 2022.
(Corressponding Author: Yan Wang.)
Digital Object Identifier no. 10.1109/TSC.2021.3062383

balancing the interests of both sides of the game is to find a QoS solution, which can not only meet the QoS demands of users, but also ensure the reasonable revenues of service providers. By analyzing users' demands for QoS, it is easy to find that different users have different QoS preferences for the service instances with the same function under different application scenarios. For example, for a SaaS application that implements a logistics distribution system, some users are concerned about its efficiency, some users are concerned about its scalability, and others are concerned about its reliability. If a service provider invests \$100 to buy the computing power and storage capacity of the platform so as to improve the reliability of the service. But for a user who only cares about the response time of the system, he is unwilling to pay for the cost of reliability. When the instance provided by the service provider fails to meet the user's QoS demands, the user will choose another service provider that can better meet his preferences. Obviously, service providers should adjust QoS based on the preferences of users, so as to obtain higher user satisfaction at lower cost. Only in this way can a service provider be in a favorable position in the competitive cloud market and ensure his/her own revenue. But in practice, it is difficult for users to accurately describe their preferences and needs. The most effective way for service providers to understand the needs of users is through the evaluations of users after using the service. Although we cannot directly derive user requirements for QoS from these evaluations, they often reflect user preferences for service performance. Assuming that user evaluations are consistent with their service performance preferences, we perform the following work.

This paper characterizes the influence of the services with different QoS on the interests of users and service providers in the competitive cloud services market, establishes the utility function of both sides of the game based on QoS. In the process of service transactions, service providers cannot predict user preferences in advance, and users cannot predict their experiences after using a service in advance, too. But, a user can choose a service provider based on the QoS of the published instance, and a service provider can dynamically decide his/her QoS based on user satisfaction. It is a dynamic game with incomplete information between service providers and users [1]. We analyze the equilibrium conditions of the game and propose a QoS evolutionary algorithm driven by the utility game for improving cost efficiency of users and resource efficiency of service providers. The contributions of this paper can be summarized as follows:

1) Regarding QoS as the focal element, this paper analyzes the conditions to reach the equilibrium state of the game between users and service providers, and introduces the reputation cost to balance current revenue and long-term revenue of a service provider.
2) Based on the theory of incomplete information evolutionary equilibrium, this paper gives the utility functions of users and service providers in the game, and establishes a utility game between two parties.
3) Based on the above analysis, this paper presents a QoS optimization algorithm to solve an optimal QoS solution, which can tradeoff user satisfaction and provider profit.

The remainder of this paper is organized as follows:

Section 2 introduces existing related work and summarizes the differences between our work and these studies.

Section 3 formalizes the utility objectives and the constraints of users and service providers in a competitive cloud service market with asymmetric information.

Section 4 gives a theoretical analysis of game model between users and service providers and establishes a utility game model in terms of user preferences and QoS.

Section 5 presents a game model driven QoS optimization algorithm to achieve the nearly optimal QoS solution.

Section 6 discusses the convergence and optimality of the proposed algorithm on numerical results and compares the results with different conditions and other optimization algorithms through the simulations, which is followed by a conclusion.

Section 7 concludes this paper.

2 Background and Related Works

In clouds, a three-tiered architecture has been formed including data center, cloud services and end users, which corresponds to infrastructure providers, service providers and users, respectively. Our work focuses on the cloud market between service providers and end users. There have been a number of studies exploiting the resource allocation and task scheduling methods to resolve the conflicting objectives of service providers and users [2]. These studies can be roughly divided into two categories: optimizing QoS to improve the user satisfaction and minimizing the cost of service to increase the revenues of service providers while meeting the users' basic QoS demands.

For services with equivalent functions, users expect to get the QoS as high as possible. Some research works improve single or multiple QoS metrics by optimizing resource scheduling to attract more users. Among these studies, the response time is considered to be one of the most important QoS. In [3], [4], the end-to-end delay of a service was minimized by a workflow scheduling algorithm while meeting the budget constraint. In the study of QoS optimization, the heuristic methods or meta-heuristic methods have also been effectively used for the scheduling problem to achieve improved service performance. In [5], the Ant Colony Optimization (ACO) was used to improve the performance of cloud service in terms of reliability, response time, cost and security. In [6], the machine learning technique was used to establish a quantitative relationship between QoS and service resources. Considering the matching relation between tasks and resources, an algorithm based on mixed game was built to allocate the most valuable resource for the tasks in [7] so that the QoS of different categories of user tasks was improved. In terms of QoS optimization of composite services, an approach for QoS-aware service composition with graph plan and fuzzy logic was proposed in [8]. The study fully considered the users' preference for service performance, employed fuzzy rules to evaluate and rank services, and then selected the optimal QoS by the GraphPlan algorithm. In [9], a complicated QoS optimization of data-intensive applications (DiA) in a hybrid cloud was studied. A DiA was modeled as a role-based collaboration system. A collaborative optimization approach via IBM ILOG CPLEX optimization package was proposed.

Although a user hopes the QoS performance is as high as possible, he won't buy it without considering its cost. Maximization of QoS and minimization of cost are the ultimate goal of users [10]. Some other studies have mainly focus on optimizing cost under QoS constraints. These studies assume that users can describe their QoS requirements in precise term. In [11], the authors presented two workflow scheduling algorithms for IaaS to minimize the execution cost of workflow while meeting the user defined deadline. In [12], a market-oriented hierarchical scheduling strategy in cloud workflow systems was proposed. In [13], the authors used a modeling framework–ROAR to choose the most optimal and cost-effective set of cloud resources to meet QoS goal of a given web application. In [14], [15], the authors maximized profit within the satisfactory level of service quality specified by the users through scheduling service requests. In [16], a resource allocation algorithm was proposed to minimize the infrastructure cost and SLA violations. In [17], a genetic algorithm for mashup creation was presented to create service mashups with achieving the optimal cost performance, which accounted for service packages and parameter transmission time saving in mashup deployment.

To investigate the trade-off between the cost of service and the expected QoS, an analytical model for the virtualization frameworks was proposed and the multi-criteria utility functions were formulated in [18]. In [19], the authors modeled the relationship between the cost and the QoS of the OSN service and designed a greedy algorithm to maximize the total cost reduction while meeting predefined QoS. But it is even harder to achieve well-compromised trade-offs, where the decision largely improves the majority of the objectives; while causing relatively small degradations to others [20]. For this, the authors in [21] presented an ant colony inspired multi-objective optimization for adaptively producing autoscaling decision that leads to a well-compromised trade-off with heavy human intervention. Some studies take the cost of service as a QoS metrics. In [22], the authors optimized both makespan and cost as a Multi-objective Optimization Problem (MOP) for the cloud environments, and then proposed an evolutionary multi-objective optimization (EMO)-based algorithm to solve the problem of workflow scheduling on IaaS platform. In [23], an evaluation framework of resource allocation strategies was proposed to conduct complex QoS queries on resource allocation instances. They enabled the tuning of parameters to improve the overall QoS through quantitative and qualitative comparisons. In order to optimize the quality of the composite services, the artificial bee colony (ABC) algorithm was improved to handle complicated multi-objective service composition and optimal selection in [24].

The study in [25] has revealed that workload and energy are also important factors that affecting the cost optimization of cloud platform. Therefore, the scheduling problem was modeled by three criteria, constituting a multi-objective function defined by the weighted summation of the execution time, cost and load in [26]. Similar work also includes SOCCER [27], QRSF [28] and QET [29]. SOCCER is a self-optimizing resource scheduling algorithm that takes energy as QoS parameter [27]. QRSF is an efficient cloud load management framework to find the best match of resource-workload pair [28]. QET is a QoS -based energy-aware task scheduling method, which minimizes the energy consumption through QoS-aware PM selection in cloud data centers [29].

In addition, as an economic model of using IT resources, the price of cloud service will directly affect users' behaviors. The trend of cloud market shows that the utilization of dynamic pricing schemes is being increased [30]. How a service provider can price a service so as to optimize his/her profit in a competitive market is a remarkable problem. In [31], the authors adopted a revenue management framework, which affected the users' needs through the dynamic price mechanism. In [32], [33], two energy-aware resource pricing schemes were designed to benefit all the stakeholders in the long run in the cloud market. In [34], a group auction model was applied to find the best matching between users and providers so as to gain benefits in terms of monetary cost and resource efficiency in the cloud market. In [35], an efficient market mechanisms to commodify resources in the integration of cloud and IoT (CoT) was proposed. The QoS optimization problem of CoT applications is transformed into the resource allocation problem to its QoS demands in the CoT trading.

From these studies, we can draw the following conclusions: (1) The improvement of QoS mainly depends on the resource scheduling scheme, which also determines the cost of service. (2) There is a conflict between multiple expected QoS in most cases. We cannot improve all of the objectives simultaneously and have to set a trade-off among them. (3) The price of service is an important factor that causes the change of service demands and affects users' choice to a great extent. (4) In the cloud market, both service providers and users strive for cost efficiency. However, the interpretations of their cost efficiency are different. Users seek for best value for money while service providers primarily aim to maximize their profits [13]. Our research will find a QoS solution that balances the objectives of users and service providers based on these publications from a technical and economic perspective. The following is a formal description of the problem to be solved in this paper.

3 Problem Formulation

We consider a simplified cloud service platform where all service providers can provide a kind of service with similar function, denoted by s.

Let SP denote the set of the service providers that can provide$s$. Formally, $SP = \{sp_1, sp_2, \ldots, sp_m\}$ where m ($m > 0$) is the number of service providers in the cloud platform. Service provider sp_i provides service $s_i (1 \leq i \leq m)$, and the set of the corresponding services is denoted by S, $S = \{s_1, s_2, \ldots, s_m\}$. Each service provider provides services to users by renting unlimited underlying resources in the data center. They can change the amount of rental resources to manage the cost and the quality of service. The QoS attributes of service s can be described by k parameters. It is denoted by $Q(s)$, which is a k-dimension vector. The execution of each service requires the invocation of multiple virtual resources. We define all kinds of virtual resource contained in the platform as the set of resources, denoted by $R = \{r_1, r_2, \ldots, r_b\}$, where $b > 1$. The performance of each service is positively related to the quantity of rental resources. Assume that the unit price of all virtual resources

is constant. Under the condition, the higher the quality of service is, the more service resources need to be invoked, and the higher the cost of service will be. Let $SC(s)$ denote the cost of service s. Therefore, we can think that $SC(s)$ is mainly determined by $Q(s)$. f_c is used to represent the mapping relationship between $Q(s)$ and $SC(s)$, then $SC(s) = f_c(Q(s))$.

Let U denote the set of users who need to request services, $U = \{u_1, u_2, \ldots, u_n\}$, where n is the number of users in U. Different users may have different demands for the QoS attributes. Let $P(u)$denote the preference of user $u$on the QoS attributes of service s, which is a k-dimension vector corresponding to $Q(s)$. $P(u)$ represents a user's different attention to various QoS attributes. Assume that all the users in U can be divided into h $(0 < h < n)$ groups according to their preferences in our previous work [36]. Let $G(u)$denote the group that user $u$belongs to. If $G(u_x) = G(u_y)$, then $P(u_x) = P(u_y)$, where $u_x, u_y \in U$, namely, the users in the same group have similar preferences. And each user's preference is stable, so a user only belongs to one group. Obviously, the user satisfaction with service is related to the quality of service and the preference of user. Let $Fc(u, s)$ denote user u's satisfaction with service $s(u \in U, s \in S)$, $Fc(u, s) = f_e(Q(s), P(u))$. $f_e(Q(s), P(u))$ is a function that depends on the variables $Q(s)$ and$P(u)$.

When a user requests a service, he/she hopes to get a service with highest quality but lowest price. But for a service provider, he/she aims to maximize his/her revenue of rental service. The revenue depends on two factors: net profit per service and the number of rental services. For the former, the service provider seeks to keep the cost of service as low as possible at the same price of service. For the latter, the service provider tends to improve user satisfaction to attract more users. Because users cannot interact with all services in the cloud market, which service a user will choose depending largely on the satisfaction of other users with similar preferences. Obviously, the objectives of users and service providers are conflicting. If the service provider provides a service with higher QoS to improve user satisfaction, it will inevitably lead to his/her higher cost and lower revenue. Otherwise, if the service provider lowers QoS to save cost, it is bound to result in a loss on user satisfaction and less service requests. From this point of view, QoS not only determines the satisfaction of a user, but also determines the revenue of a service provider in the service transaction. Therefore, finding a compromised QoS that meets the requirements of both parties is the key to resolve conflicts between users and service providers. For service providersp_i, the solution to the compromised QoS for the users in $G(u_x)$can be formalized as follows.

$$\max Fc(u_x, s_i) = f_e(P(u_x), Q(s_i)) \tag{1}$$

$$\min SC(s_i) = f_c(Q(s_i)) \tag{2}$$

subject to

$$u_x \in U, s_i \in S \tag{3}$$

$$SC(s_i) < \Pr(s_i) \tag{4}$$

$$R(sp_i, G(u_x)) > \underset{i \neq j}{Max} R(sp_j, G(u_x)) \tag{5}$$

where

$$\forall u_z \in U, \forall sp_c \in SP \cup s_c \in S$$
$$R(sp_c, G(u_z)) = \sum_{u_y \in O(G(u_z), s_c)} Fc(u_y, s_c) / |O(G(u_z), s_c)| \tag{6}$$

$$\text{if } |O(G(u_z), s_c)| = 0, R(sp_c, G(u_z)) = 0 \tag{7}$$

Equation (1) represents the objective of u_x, while Equation (2) represents the objective of sp_i. For u_x, the maximization of $Fc(u_x, s_i)$ means that he can get the best value for the payment of the service. For sp_i, the minimization of $SC(s_i)$ means that he can obtain higher revenue per service. Constraint (3) states the scope of entities participating in service transactions. Constraint (4) states the condition of service transactions, namely, the cost of service $SC(s_i)$ must be lower than the price of service $\Pr(s_i)$. Constraint (5) expresses the condition of user u_x choosing services_i, where $R(sp_i, G(u_x))$ represents the reputation of service provider sp_i for the users in $G(u_x)$. The essence of reputation is the representation of user satisfaction. For u_x, $R(sp_i, G(u_x))$ is the average satisfaction of the users in $G(u_x)$ with the service provided by sp_i. Whether or not s_i is selected by u_x depends on $R(sp_i, G(u_x))$. s_i can be selected by u_x if and only if the reputation of sp_i is the highest among all the service providers. For sp_i, Equation (5) is the guarantee of the number of user requests. Equations (6) and (7) state the evaluation method of the reputation. In (6), $O(G(u_z), s_c)$ represents the set of the users in $G(u_z)$ that have selected service s_c, $|O(G(u_z), s_c)|$ is for the number of users in $O(G(u_z), s_c)$. So $R(sp_c, G(u_z))$ is the average value of historic user satisfactions with s_c for the users in $G(u_z)$. We regard $R(sp_c, G(u_z))$ as the reputation of sp_c for $G(u_z)$. When $|O(G(u_z), s_c)| = 0$, namely, the users in $G(u_z)$ have never selected s_c, $R(sp_c, G(u_z)) = 0$.

This promblem model exists in a competitive market with asymmetric information. Therefore, it contains some uncertain relationships, which is intractable for practical instances. Below, we propose a QoS optimization method based on game theory to balance the interests of users and service providers.

4. Utility Game Model Between Users and Service Providers

In the actual cloud service environment, there is a serious information asymmetry. On one hand, a user doesn't know how well does the QoS provided by a service provider. On the other hand, a service provider is unable to accurately grasp users' demands. Therefore, there must be a game relationship between them. The behaviors of service providers and users are interacting and influencing each other. QoS is the focus of users and service providers. The key factor in the game depends on whether a service is equivalent for both parties or not. For this, we will analyze the characteristics of the game between users and service providers in the cloud service market, establish a utility game model between them, and discuss the equilibrium solution of the game. In order to make it easier to read, we first give all the symbols and their meanings involved in the paper in Table 1.

TABLE 1
The Symbol Description

Symbol	Meaning
s	Service
u	User
SP	a set of service providers
S	a set of services
$Q(s)$	the quality of service s
R	a set of virtual resources
$SC(s)$	the cost of service s
$P(u)$	the preference of user u
$Pr(s)$	the price of service s
$G(u)$	the group of user u
$Fc(u,s)$	the satisfaction of user u with service s
$C^R(s)$	the resource cost matrix of service s
$C^B(s)$	the basic service charge of service s
$D(u,s)$	the QoS recognition deviation of user u on service s
$V(u,s)$	the user perceived value of user u on service s
$Q^{uMin}(s)$	the acceptable lowest QoS of services s for user u
$U_c(u,s)$	the user utility of service s for user u
$SRC(s,G(u))$	the reputation cost of service s for gaining the users' satisfaction in group $G(u)$
$U_p(s,u)$	the utility of service s for service provider p when providing services to users in group $G(u)$

4.1 Game Analysis Based on QoS

A complete game should contain five aspects: the participants of the game, the information of the game, the set of all behaviors or strategies that can be selected by the players, the order of the game and the benefits of the players. In the game discussed in this paper, the participants are SaaS providers and end users in the cloud service market. In the game, service providers can evaluate the cost of service based on the used resources, know the price of other similar services in the platform, and gain users' scores on the used services. Users know which service instance can provide the required service functions and give their scores on the services they have used according to their preferences. For a service provider, the strategies he/she can adopt include two aspects: the price of service and QoS, denoted by {Pr(s), Q(s)}. For a user, the actions he/she can take are whether to choose a service instance provided by the service provider or not. Once a service provider determines the quality of service, users will decide whether to choose the service according to their demands. For every possible decision of both sides of the game, there should be a result representing the gain and loss of each player under the decision, which is called the utility function.

To analyze the game, we need to consider the following questions: 1) How to represent a user's preference -$P(u)$; 2) How to measure the relationship between the QoS and the cost of rented resource -f_c; 3) How to measure the relationship between the QoS and the user satisfaction - f_e; 4) How to evaluate the impact of user satisfaction on the revenue of a service provider. In what follows, we will give the formal definitions of these questions.

Firstly, QoS is a concrete representation of service performance, covering multiple metrics of a service. It can be represented as a multidimensional vector. Because different QoS parameters have different metric and scope, we standardize them to describe a service.

Definition 1 (QoS) [37]. *The quality of service is described as a normalized k-dimension vector. Formally,*

$$Q(s) = (q_1(s), q_2(s), \ldots, q_k(s)), \tag{8}$$

where

$$q_{\mathrm{i}}(\mathrm{s}) = \begin{cases} (q_{\mathrm{i}}^{Max}(s) - q_{\mathrm{i}}^{A}(\mathrm{s}))/(q_{\mathrm{i}}^{Max}(s) - q_{\mathrm{i}}^{Min}(s)) & q_i \in Q_N \\ (q_{\mathrm{i}}^{A}(\mathrm{s}) - q_{\mathrm{i}}^{Min}(s))/(q_{\mathrm{i}}^{Max}(s) - q_{\mathrm{i}}^{Min}(s)) & q_{\mathrm{i}} \in Q_p \end{cases}. \tag{9}$$

In (9), $q_{\mathrm{i}}^{Max}(s)$, $q_{\mathrm{i}}^{Min}(s)$ and $q_{\mathrm{i}}^{A}(\mathrm{s})$ are for the maximum value, the minimum value and the actual value of ith attribute of service s, respectively. All the attributes are divided into two categories: positive correlation and negative correlation. Q_p is a set of positive relevant attributes; Q_N is a set of negative relevant attributes. For the former, a higher value is better, such as throughput, reliability and availability; for the latter, lower is better, such as response time. Corresponding to this, the definition of user preference is given as below.

Definition 2 (User Preference) *Under the constraint of service payment, the users pay more attention to one or serval QoS parameters, that is, the users are willing to spend more money to invest in the improvement of the QoS parameters with high degree of attention. Formally,*

$$P(u) = \{p_1(u), p_2(u), \ldots p_k(u)\}, \tag{10}$$

where $p_{\mathrm{i}}(u)$ represents the user's preference for the ith QoS attribute. For comparison, we limit $P(u)$ subject to $\sum_{i=1}^{k} p_i(u) = 1$.

In cloud pricing schemes, the cost of service s is determined by the amount or running time of underlying virtual resources used by s. The cost of service $SC(s)$is different for different $Q(s)$. To simplify the problem, assume that $SC(s)$ varies proportionally to $Q(s)$, furthermore, $Q(s)$ varies proportionally to the amount of used resources in R. For a type of service, the cost per unit for improving any QoS parameter is fixed without changing other QoS parameters. Below, give the definitions of resource cost and service cost.

Definition 3 (Resource Cost Matrix). *For b-class virtual resources supplied by the cloud, $R = \{r_1, r_2, \ldots, r_b\}$, the cost of used resources for improving all QoS parameters of service s is denoted as $\mathrm{C^R}(s)$. Formally,*

$$C^R(s) = \begin{bmatrix} c_{11} & c_{12} & \ldots & c_{1b} \\ c_{21} & c_{22} & \ldots & c_{2b} \\ \ldots & \ldots & \ldots & \ldots \\ c_{k1} & c_{k2} & \ldots & c_{kb} \end{bmatrix}, \tag{11}$$

where c_{ij} represents the payment of the used resource r_j in order to improve $q_i(s)$ from $q_{\mathrm{i}}^{Min}(s)$ to $q_{\mathrm{i}}^{Max}(s)$ while other QoS attributes remain unchanged.

The cost of service is based on many factors, including the cost of used resources, management fees and implicit fee clauses in the SLA. The cost of service considered in this paper is limited to the cost of the used resources, which can be divided into two parts: the minimum fixed cost for a service and the cost of improving the performance of the service. The definition of service cost is given as follows.

Definition 4 (Service Cost). *$SC(s)$ is defined as the payment for the used resources when a service provider provides a service with $Q(s)$. Formally,*

$$SC(s) = C^B(s) + Q(s) \times C_Q^R(s), \tag{12}$$

where C^B is the basic service charge per request, that is, the cost of providing $Q(s) = (q_1^{Min}(s), q_2^{Min}(s), \ldots, q_k^{Min}(s))$. $C_{\mathrm{Q}}^{\mathrm{R}}(s) = (c^1, c^2, \ldots, c^{\mathrm{k}})^T$, $c^i = \sum_{j=1}^{b} c_{ij}$, represents the payment for all kinds of rented resources when $q_i(\mathrm{s})$ is improved from $q_i^{Min}(\mathrm{s})$ to $q_i^{Max}(\mathrm{s})$ under with other QoS parameters unchanged.

For a user with specific QoS preferences, when the QoS provided by a service provider is different from his/her demands, he/she will think that the value of the service is less than the price he/she pay; otherwise, he/she will think that the service is good value for money. We call the value of a service to a user as user perceived value. The relevant concepts are given below.

Definition 5 (QoS Recognition Deviation). *Assuming that the price of service s is p, the difference between the expected QoS of user u and the actual QoS is called the user's QoS recognition deviation. Formally,*

$$D(u, s) = \sum_{i=1}^{k} p_i(u) \times (q_i^u(s)/q_i(s)), \tag{13}$$

where $q_i^u(s)$ represents the value of the kth parameter in Q(s) expected by user u when $\mathrm{Pr}(s) = p$.

Obviously, when $q_i(s) < q_i^u(s)$for each i ($i \in [1, k]$), user u believe that service s cannot fully meet his/her demands for service performance.

Definition 6 (User Perceived Value). *A user's subjective cognition of a service's value according to his/her own preference is called user perceived value. Formally,*

$$\mathrm{V(u,s)} = \begin{cases} SC(s)/D(u,s) & Q(s) \geq \mathrm{Q^{uMin}}(s) \\ 0 & otherwise \end{cases}, \tag{14}$$

where $\mathrm{Q^{uMin}}(s) = (q_1^{u\mathrm{Min}}(s), q_2^{u\mathrm{Min}}(s) \ldots, q_k^{u\mathrm{Min}}(s))$ represents the acceptable lowest QoS of service s for user u.

When $Q(s) < \mathrm{Q^{uMin}}(s)$, user u thinks service s is worthless. For example, user u asks for the response time of service s up to 2 seconds. But the response time of service s is more than 2 seconds. In this case, the user is not willing to buy such a service. And a user's recognition for the value of a service is directly proportional to the user's QoS recognition deviation. When $D(u, s) = 1$, the user thinks the value of the service is conistent with its price. If $D(u, s) > 1$, the user thinks the value of the service is less than its actual price.

Below, we will discuss the quantitative relationship between $P(u)$, $Q(s)$, $D(u, s)$ and $\mathrm{V(u,s)}$. It is assumed that a SaaS service can be described by two QoS attributes: response time and security, that is, $Q(s) = (\mathrm{q_1(s), q_2(s)})$. When the service is executed, it needs to call three kinds of resources in the cloud platform: CPU, storage and network, namely $R = \{\mathrm{r_1, r_2, r_3}\}$. Set $\mathrm{C^R}(s) = (3, 2, 4; 4, 5, 3)$, $C^B(s) = 2$, $P(u) = (0.8, 0.2)$ and $\mathrm{Q^{uMin}}(s) = (0.3, 0.1)$. The quantitative relationship between $D(u, s)$ and $Q(s)$ is shown in Fig. 1. As shown in

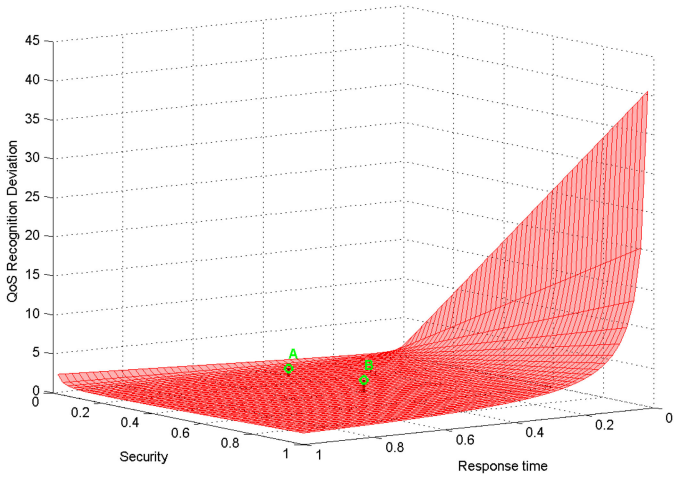


Fig. 1 The quantitative relationship between $Q(s)$ and $D(u, s)$.

point A in the figure, when $D(u, s) = 1$, $Q(s) = (0.4, 0.1)$. At this time, we can see that the QoS is consistent with the user's preferences, that is, the user pays more attention to the response time of service$s$. When Q(s) is inconsistent with user u's preference, there is a certain deviation between the expected quality of service of user u and the quality of service provided by a service provider. For example, as shown in point B in the figure, when $Q(s) = (0.4, 0.4)$, $D(u, s) > 1$. Even though a service provider spends more money to provide service s, the QoS in point B is no different from that in point A for user u. The user is reluctant to pay for the extra costs. If user u purchases the service at this cost, the expected $Q^u(s)=$ (0.7, 0.175). In this case, with the increase of the deviation, the user's QoS recognition deviation for the service becomes higher and higher. Specially, when a service provider only focuses on the security of the service and ignores the response time of the service, $D(u, s)$ reaches its maximum value.

Similarly, suppose that user u thinks the lowest acceptable QoS for a service $\mathrm{Q^{uMin}}(s) = (0.2, 0)$. The relationship between $Q(s)$, $\mathrm{Pr}(s)$ and $V(u, s)$ is shown in Fig. 2. The top surface of the figure corresponds to the cost of service of different QoS, while the bottom surface corresponds to the user perceived value of different QoS (Without loss of generality, we assume that the price of service s is proportional to its cost, and the ratio equals 1). From Fig. 2, we can see that when the QoS is inconsistent with the user's preference, then $V(u, s) < \mathrm{Pr}(s)$. For example, when $Q(s) = (0.7, 0.8)$, $\mathrm{Pr}(s) =$ 15.9, $V(u, s) = 9.96$ (See point A and B in the figure). In this case, the user is only willing to purchase the service according to the perceived value of the service. In turn, the price offered by users will affect the distribution of service quality provided by service providers. If the cost of service is higher than the perceived value of users, service providers should adjust the quality of service to accommodate the user's bid. In the case that the service provider does not know the user's preference, it will inevitably lead to the decline of the overall service quality. In turn, it will result in further reductions in user acceptable price. Therefore, there must be a game relationship between service providers and users, prompting service providers to adjust the quality of service according to user preferences so as to improve users' satisfaction with the services that have the same value.

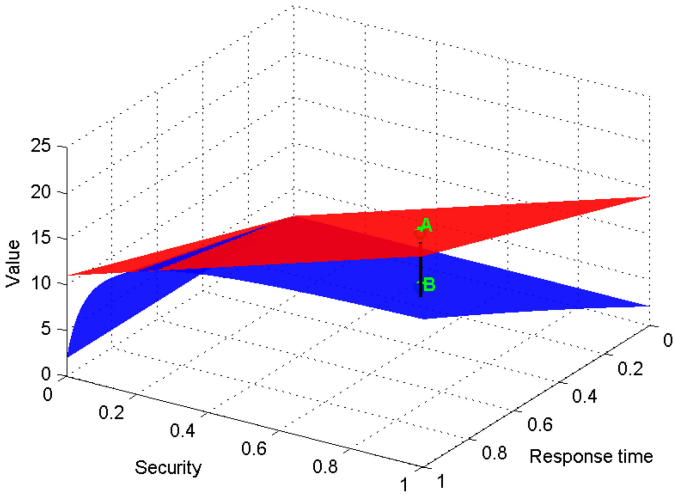


Fig. 2 The quantitative relationship between $Q(s)$, $\Pr(s)$ and $V(u,s)$.

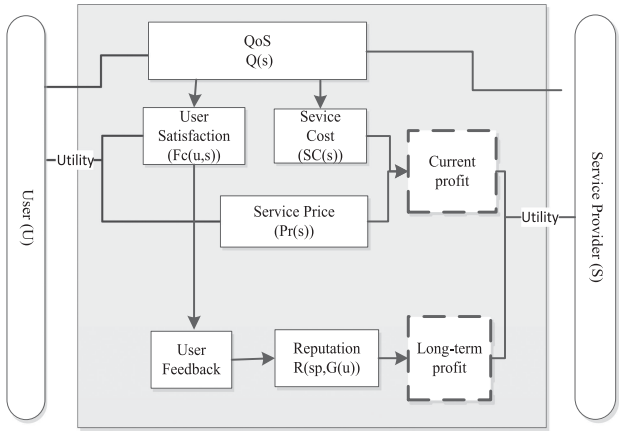


Fig. 3 Game based on utility between users and service providers.

4.2 Game Model Based on Reputation Cost

The nature of cloud service transactions is an economic activity, so the competition and interaction in the cloud service market are similar to the free competitive market in economics. In the competitive market environment, the behavior of each economic entity is subject to the behavior of other participating entities [38]. While users and service providers are pursuing their own interests, they inevitably have conflicts with each other. Therefore, there must be a game relationship between users and service providers. Most of the game models are based on the rationality of the players. But in reality, it is difficult for users and service providers to be completely rational. Due to the information asymmetry in the cloud service market, the rational limitations of users and service providers are very obvious. Therefore, to ensure the theoretical and practical value of game analysis, it is necessary to analyze the game between limited rational players. Evolutionary game theory is usually used to study such problems, and the key to evolutionary game analysis is to determine the mechanism of learning and strategy adjustment. In this paper, we will establish a long-term game model between the users with similar preferences and service providers, and then study the process of game in which a service provider can realize the maximization of user satisfaction and his/her own interests by mastering user information. Next, we focus on the analysis of the payoff of users and service providers in the long-term game process.

The payoffs of both parties are their respective utilities, as shown in Fig. 3. The utility of the user is associated with $Fc(u,s)$ and $\Pr(s)$. The lower the price is and the higher the user satisfaction is, the greater his/her utility is. The revenue of service provider is decomposed into the current profit and the long-term profit. The current profit, denoted by $CR(s)$, is net profit of one deal, that is, $CR(s) = \Pr(s) - SC(s)$. But one-time profit is not an ultimate goal of a rational service provider. Each service provider will give more consideration for his long-term profit, namely, increasing sale numbers. Whether or not a service with the given QoS can be recognized by its provider and user as being of equal value will become the key to the equilibrium of game.

As mentioned above, if the service provider provides a service inconsistent with user preferences, the service experience of users will be poor. In fact, such phenomenons of information asymmetry are widespread in the cloud market. In such a circumstance, when a user requests an unknown service, he chooses it based on the reputation of service providers or the recommendations from other similar users. After one deal is over, each user will submit his satisfaction feedback to his service provider. In the end, a large number of user feedback forms the reputation of the service provider, namely, $R(sp, G(u))$. Good reputation is the guarantee of the long-term profit of the service provider.

But due to different preferences, different users sometimes give different feedbacks on the service with the same quality. The clustering problem of users with similar preferences has been explored in our previous work, and we call it cloud community [36]. In [39], an algorithm was proposed to computes the scores of QoS of web services within a community. Inspired by these studies, we abstract a cloud community as a player in the cloud market, in which each individual has the same interest and consistent behavior. Thus, the game between users and service providers is translated to be a game between a cloud community and a service provider. In this case, a user's choice behavior depends on the historical user feedbacks from his community. If a service provider provides a service that is not in line with the users' preference in the community, it will result in poor user feedback and loss of reputation. In this sense, the game between user communities and service providers is not a one-time game, but a multi-stage observable game of incomplete information. At each stage of the game, a user makes his/her decision by the user feedbacks in previous stages and a service provider designs a QoS solution based on the user feedbacks. The user is only interested in the utility brought by a single service, not caring about how many users have bought the service. But a service provider seeks the total profits from the provided services, not just one service's profit. Therefore, the game between cloud communities and service providers is a multi-stage dynamic game.

Based on the above discussion, we give the utility functions of service providers and users in the process of cloud service transaction. The utility of user is used to describe the value of service for a user. If QoS is equal to or higher than the QoS expected by the user, the user thinks that the service has a good value for money; otherwise the user thinks that the service is not worth his/her payment. The

bigger the difference between expected QoS and actual QoS is, the lower the value of service is, that is, the lower the utility of user is. So the value of service for a user is determined by two factors: the price of service and the difference between the QoS expected by a user and the QoS provided by a service provider. Assume that the net profit rate of service is the same in the cloud market, that is, the price of service is the same when the cost of service is the same. In this case, $\Pr(s) = SC(s)/(1-R)$, where $R(0 < \mathrm{R} < 1)$ is for the profit rate. The utility of user is defined as follows.

Definition 7 (User Utility Function). *The user utility of a service is defined as the difference between the actual price and the value appraised by the user. Formally,*

$$U_c(u,s) = \min(\Pr(s) * 1/D(u,s), \Pr(s)) - \Pr(s). \tag{15}$$

When $D(u,s) \leq 1, U_c(u,s) = 0$, the user is fully satisfied with the service.

For any service provider sp, his/her profit from service s is the difference between the price of service $\Pr(s)$ and the cost of service $SC(s)$ in each service transaction. It can be seen through the above analysis that the high user satisfaction can bring a good reputation, and a good reputation means that the service provider can obtain more user requests and more profits in the future. Therefore each service provider cannot ignore the user satisfaction with the service he/she provides. From the long-term interests, each service provider is bound to pursuit a higher user satisfaction. Howerver, building a good reputation cost a lot more. In order to achieve better satisfaction, service providers have to sacrifice some of their own current profit to improve the quality of service. In the paper, the definition of reputation cost, denoted by $SRC(s, G(u))$, is presented to balance the current profit and the user satisfaction of a service provider. If the service provider just pursues the net profit per service, it will decrease the user satisfaction, resulting in the loss of the provider's reputation and a rapid decline in the number of user requests. If the service provider wants to increase the number of user requests, he/she needs to improve $Q(s)$ to increase the user satisfaction. In general, the reputation of service provider is made to increase slowly/decrease rapidly by the influence of user satisfaction. So $SRC(s, G(u))$ should grow or decline in a nonlinear tendency along with user satisfaction. In view of this idea, the nonlinear transforming parameter function based on the nonlinear hyperbolic tangent function, $y = \tanh Ax$, is used to quantify $SRC(s, G(u))$, where $A(A \geq 1)$ is constant for the curve gradient. In the function, the step size of $SRC(s, G(u))$ is small when the user satisfaction is high, otherwise is big. The proposed model is based on our previous assumptions. To our knowledge, such behavior has not been studied. Below, we give the definition of reputation cost.

Definition 8 (Reputation Cost). *The reputation cost reprensents spending money on user satisfaction of a service provider to gain long-term profit. Formally,*

$$\begin{aligned} SRC(s, G(u)) = (1 - \tanh(A \times 1/D(u,s))) \\ \times\ (\Pr(s) - SC(s)) \end{aligned} \tag{16}$$

The reputation cost has a constraint on the behaviors of a service provider to provide poor services. At the same time, it also creates conditions for service providers to gain higher total profit. For all this, the reputation cost is introduced into the utility function of service provider in a single game. The utility of service provider is defined as follows.

Definition 9 (Service Provider Utility Function). *The utility of service provider is the current net profits of service subtracting the cost for accumulating reputation in one-time transaction. Formally,*

$$U_p(s,u) = \Pr(s) - SC(s) - SRC(s, G(u)), \tag{17}$$

where Pr(s) is for the price of service s, SC(s) is for the cost of services.

For service providers, the utility of a service is its revenue, that is, its price minus its cost. But, if a service provider only pursues his/her revenue and ignores the preference and satisfaction of users, it will inevitably lead to the reduction of users' choice. In this formula, we introduce $SRC(s, G(u))$ to constrain the service provider's adjustment strategy for Q(s), so that the service provider can constantly improve user satisfaction through sacrificing some revenue, and achieve the goal of balancing his/her short-term profit and long-term profit.

5 Utility Game Driven QoS Optimization

Obviously, the game process between the user and the service provider mentioned above is a dynamic game based on incomplete information. The key to solve our problem in this paper is to find the Nash equilibrium solution of this game. Evolutionary game theory is a kind of modeling method used to solve the dynamic game of incomplete information. This theory holds that all the game players in reality are finite rational individuals. Each individual can't directly determine his/her own optimal strategy, which needs to dynamically find the optimal behavior through the imitation, learning and even mutation of the individual. In the process, "successful" strategies are replicated to achieve more "satisfactory" utility [40]. For the problem of equilibrium solution for dynamic game, Hirshleifer organically combined the static concepts and dynamic processes in evolutionary game theory, and gave the concept of Evolutionary Equilibrium (EE). EE is defined as a local asymptotically stable equilibrium point. From any small neighborhood of the equilibrium point, the trajectory will eventually evolve to this point. When the equilibrium state is reached, the strategy adopted by the participants is the Evolutionary Stability Strategy (ESS). At this time, every individual in the whole population adopts this strategy, and under the action of natural selection, there is no mutation which can violate this strategy [41]. Our ultimate objective is to solve a QoS solution to reach EE state of the game between a cloud community and service providers. The user preferences oriented EES is beneficial to the efficiency of service cost, and helps to realize a win-win situation between service providers and users. Base on above theory, we designed a QoS optimization algorithm of cloud services based on the utility functions of both players to automatically find the optimal

QoS solution to reach EE state. In the process of QoS optimization, the problems that which QoS parameter should be improved firstly and how to find a tradeoff among all the QoS parameters are the main considerations.

ALGORITHM 1: QoS optimization based on utility game (QoSUG)

Input: $P(\mathrm{u})$, $C^R(\mathrm{s})$, $C^B(\mathrm{s})$, R
Output: $Q(s)$ (the optimal QoS solution for G(u))
Procedure UtilforUser($P(\mathrm{u})$, $Q(s)$, $\mathrm{Pr}(s)$, CE^R, $SC(s)$)
{
$P(u)^T \leftarrow reshape(P(u), 1, k)$
$UB = CE^R \times P(u)^T$
$UBE = SC(s)/UB$
$[Ps(u)\ pind] \leftarrow sort(P(u)^T, 'descend')$
for i = 1:k
$q_i^{\mathrm{u}}(s) \leftarrow CompQx(Ps(u), UBE, CE^R)$
end for
$Q^{\mathrm{u}}(s) \leftarrow (q_1^{\mathrm{u}}(s), q_2^{\mathrm{u}}(s), \ldots q_k^{\mathrm{u}}(s))$
$D(u,s) = CompDu(Q^{\mathrm{u}}(s), Q(s), P(u)^T)$
if $D(u,s) < 1$ then
$U_c(u,s) = \mathrm{Pr}(s) * (D(u,s) - 1)$
else
$U_c(u,s) = 0$
end if
return($U_c(u,s)$, $D(u,s)$)}
Procedure UtilforService($\mathrm{Pr}(s)$, $SC(s)$, $D(u,s)$)
{
$\theta = \mathrm{A} * (1 - \mathrm{D(u,s)})$
$\beta = 1 - \frac{e^{\theta} - \mathrm{e}^{-\theta}}{e^{\theta} + \mathrm{e}^{-\theta}}$
$SRC(\mathrm{s}, \mathrm{G(u)}) = \beta * (\mathrm{Pr\,(s)} - \mathrm{SC(s)})$
$U_p(s,u) = \mathrm{Pr}(s) - SC(s) - SRC(s, G(u))$
return($U_p(s,u)$)
}
$ChromQoS \leftarrow GenQoS(NIND)$ // Random distribution for QoS
for i = 1:k
$\mathrm{ce}_n = \sum_{j=1}^{b} c_{ij}$
end for
$\mathrm{CE^R} \leftarrow (\mathrm{ce}_1, \mathrm{ce}_2, \ldots \mathrm{ce_k})$
Do
{
for n = 1:NIND
$SC(\mathrm{s}_n) = \mathrm{C^B(s)} + ChromQoS(n) \times \mathrm{CE^R}$// the cost of service
$\mathrm{Pr}(s_n) = SC(s_n)/(1 - R)$ // the price of service
$[U_c(u, s_n)\ D(u, s_n)] = \mathrm{UtilforUser}(P(u), ChromQoS(n), \mathrm{Pr}(s_n), CE^R, SC(s_n))$
// compute the utility of user
$U_p(s_n, u) = \mathrm{UtilforService}(\mathrm{Pr}(s_n), SC(s_n), D(u, s_n))$
// compute the utility of service provider
$ObjV = \alpha_1 * U_c(u,s) + \alpha_2 * U_p(s,u)$//compute the objective function
end for
$ChromQoSE \leftarrow EncodingQoS(ChromQoS)$
//encode every service as a string of binary code.
$[ObjvS\ ind] \leftarrow sort(ObjV, 'ascend')$
$FintV \leftarrow Fitness(Ind, NIND)$
$SelChrQoS \leftarrow SelectQoS(FintV, ChromQoSE)$
// finish the select operation
$ChildChrQoS \leftarrow CrossoverQoS(SelChrQoS)$
//crossover the selected individuals by linkage learning techniques and guided method [42]

$$p_m = \begin{cases} \frac{p_{m\max} - p_{m\min}}{1 + \exp(2 * \frac{ObjV - ObjV_{avg}}{ObjV_{\max} - ObjV_{avg}} - 1)} + p_{m\min} & ObjV > ObjV_{avg} \\ P_{m\min} & ObjV < ObjV_{avg} \end{cases}$$

// compute the adaptive mutation operation
$ChildChrQoS \leftarrow MutaionQoS(ChildChrQoS, p_m)$
// mutate the individuals with the probability of p_m
$ChromQoSD \leftarrow DecodingQoS(ChildChrQoS)$
// check with the QoS constraints, retain or discard
$ChromQoS \leftarrow \mathrm{Regeneration}(ChromQoSD, ChromQoS)$
// replace children of the worst fitness with the best one
} // iterative solution
While (!$Unchange(Max_{ChromQ(s)}\ ObjV(ChromQoS), MGen)$)
// until the best individual remains unchanged
$Q(s) \leftarrow Max_{ChromQ(s)}\ ObjV(ChromQoS)$ // Selecting the best solution

In order to automatically find a QoS solution to reach the EE state of the game between users and service providers, the interests of both parties are taken into consideration, and a weighted optimization method is used to transform the multi-objective problem into the single-objective problem. By this way, it is convenient to flexibly set primary optimization objective according to the actual situation of the cloud service market, while taking into account the demands of secondary objective. We can adjust the trade-off to achieve the balance between supply and demand of the cloud service market so as to promote the benign competition of service providers in the cloud platform. For example, when the supply exceeds the demand in the cloud service market, user utility is the primary optimization goal; otherwise, the weight of service provider utility should be appropriately increased. So an improved version of GA (genetic algorithm) is presented to find the exact or approximate solution to the problem of QoS optimization. GA is a search technique inspired by evolutionary biology. It uses an optimizing method with stochastic probability, automatically conducting the search space and the optimization direction through the fitness function. In GA, every solution is represented with a string, also known as a chromosome. Through three basic GA operations, i.e., selection, crossover and mutation, imitate the process of biology evolution with genetic choice and natural elimination. When the stopping condition is met, the chromosome with the best fitness value is the near-optimal solution in the search space. In the paper, the vector Q(s) is encoded as chromosome, and then the chromosome with the best fitness value is generated under the constraint of resource cost matrix $\mathrm{C^R}(s)$ and the price of service $\mathrm{Pr}(s)$. We combine the two utility functions of users and service providers and give a weight to each utility function to set up a total objective function. The objective function is formally as follows:

$$ObjV = \alpha_1 * U_c(u,s) + \alpha_2 * U_p(s,u)(\alpha_1 + \alpha_2 = 1), \tag{18}$$

When$\alpha_1 = 1$, the optimization goal centers on user utility and a QoS solution is found to maximize the user's satisfaction. When $\alpha_2 = 1$, the optimization goal centers on service provider utility and a QoS solution is found to maximize the service providers' profits. Otherwise, the objective

TABLE 2
Settings of Relevant Parameters

Parameter	Value	Meaning of the parameter
C^R	[19535; 81 2312; 371 89; 79811]	Resource cost matrix
$P(u)$	(0.4; 0.3; 0.2; 0.1)	User QoS preference
C^B	0	Service charge per request
$\Pr(s)$	100	Service price
R	40%	Mean of profit rate
GGAP	0.95	Selection operator
p_x	0.7	Crossover operator
p_m	0.01	Mutation operator
CN	30	Number of population

function will solve the optimal QoS from the perspective of the balance of both parties.

The QoS optimization algorithm is depicted in Algorithm 1, which translates the problem of QoS optimization to the survival of the fittest chromosome. When the objective function of the optimal individual in the population reaches the maximum value of individuals in all the generations and its value remains unchanged after consecutive generations, the algorithm eventually converges to a QoS solution which maximizes the objective function.

6 Experiments and Analysis

6.1 Experiment Settings

In this section, we will analyze the process of QoS optimization based on utility game between users and service providers. In order to simplify the problem and protect the experimental results from being affected by other uncertain factors, assume that the users can give their feedbacks on a service according to their preferences when they have already used the service in the experiments. The experiments are divided into four groups. One for analyzing the convergence process and optimization results when the objective functions are different, one for analyzing the influence of $SRC(s, G(u))$ on the process of QoS optimization, one for comparing the optimization results between different $P(u)$, and the last one for comparing our algorithm and others QoS optimization algorithms. The settings of relevant parameters in the experiments are shown in Table 2.

6.2 Experimental Results

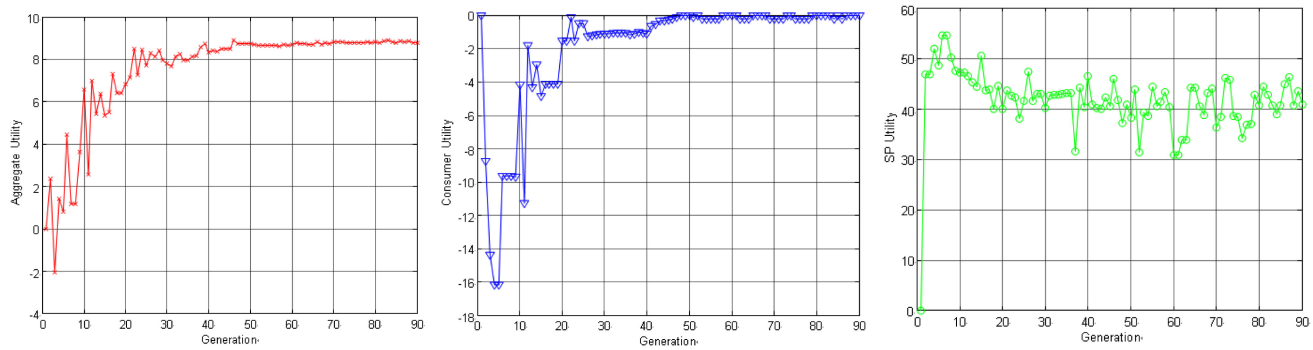
As explained in the parameter settings for the optimization algorithm, we adjust the weights of utility ($U_c(u, s)$ and $U_p(s, u)$) in $ObjV$ to analyze the convergence speed and $Q(s)$ after convergence. The experimental results are shown in Fig. 4 and Table 3.

It can be seen from Table 3 that the results of QoS optimization are different for users with similar preferences when the weights of the objective function are different. When $\alpha_1 = 0.7$ and $\alpha_2 = 0.3$, after 90 iterations, the process of QoS optimization reaches its convergence. After convergence, the best individual $Q(s) = (0.8301, 0.6254, 0.3978, 0.1862)$, the corresponding user utility and service provider (SP) utility are -0.2653 and 28.3343, respectively. In the evolutionary result, the ratio of QoS parameters of the best individual is approximately equal to the ratio of user preference, namely, $q_i(s)/q_j(s) = p_i(u)/p_j(u)(i, j \in [1, k])$. And the user utility is approximately equal to 0, which indicates that the value of service s for user u is equal to his payment for the service. When$\alpha_1 = 0.3$ and $\alpha_2 = 0.7$, after 205 iterations, the QoS optimization reaches its convergence. After convergence, the best individual $Q(s) = (0.7827, 0.4954, 0.0038, 0.0021)$, the corresponding user utility and SP utility are -30.2577 and 53.1302, respectively. In the evolutionary result, the first two QoS parameters, to which the users pay more attention, are the main concerns for the service provider. So the profit of service provider has a significant growth, which beyond the average profit rate of the market, achieving 53 percent. But at the same time, the user satisfaction has declined by 30 percent than that of the previous $ObjV$. It is very obvious that the increasing rate of SP utility brought by the optimized QoS solution through the algorithm is higher than the declining rate of user satisfaction, which maximizes the cost effectiveness of service provider. When $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$, the equilibrium of users and service providers comes true through sacrificing their respective minority utility. Seen from Fig. 4, for the optimization process focusing on one-side utility, the convergence process of aggregate utility is consistent with that of the focused side. The utility of the focused side constantly increases with the evolutional generations, while the convergence process of the other side has some unstable oscillations, as shown in Figs. 4a and 4c. The optimization process for balancing both sides tends to be stable in the slow growth with small oscillations. Further, due to reputation cost being considered in SP utility, all the SP-oriented convergence is slower than the user-oriented convergence.

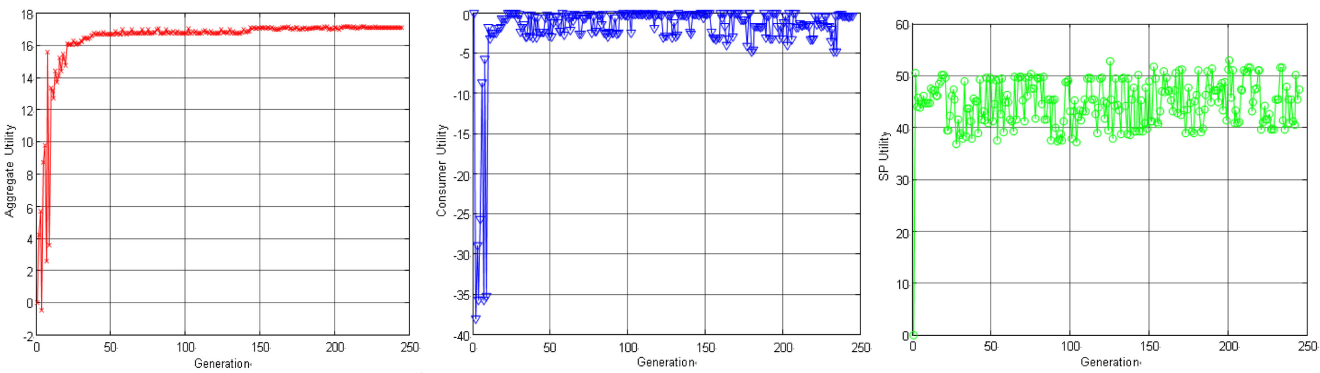
In the second experiment, how curve gradient A affects the convergence speed of the algorithm is analyzed. Set $\alpha_1 = 0$ and $\alpha_2 = 1$, other parameters are shown in Table 2. For each value of A in Table 4, calculate the mean of multiple optimization results. It can be seen from Table 4 that the bigger the value of A is, the steeper the reputation curve is, namely, the greater the influence of user satisfaction on reputation cost is. So the small value of A will result in slow evolutionary process and poor convergence effect, even premature. With the increase of the value of A, the reputation cost becomes more sensitive to the change of user satisfaction, and the convergence effect of the algorithm is gradually strengthened. When $A = 2.5$, the convergence speed and convergence effect of the algorithm are preferable. After that, when the value of A continue to increase, the

TABLE 3
The Experimental Results in the First Experment

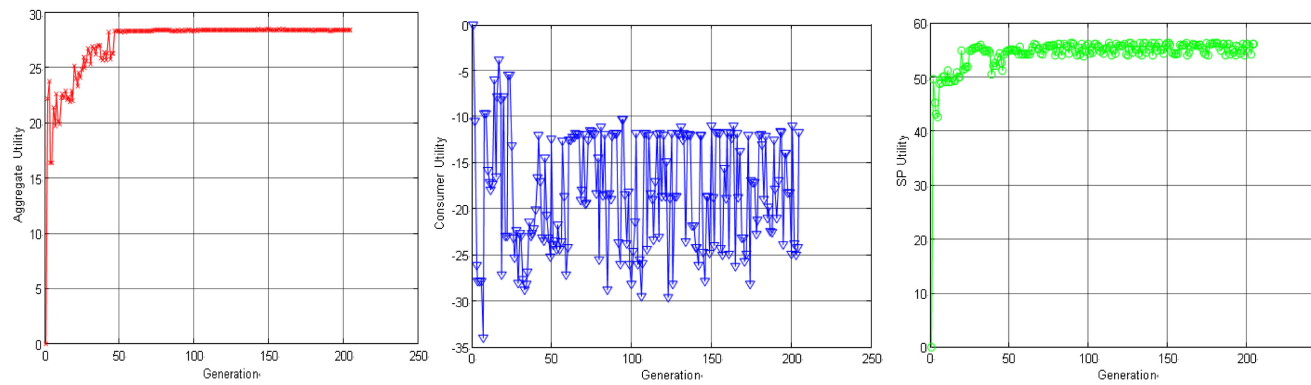
Objective Function	QoS after convergence				Aggregate utility	User Utility	SP utility	Convergence generation
$\alpha_1 = 0.7, \alpha_2 = 0.3$	0.8301	0.6254	0.3978	0.1862	8.7381	-0.2653	28.3343	90
$\alpha_1 = 0.5, \alpha_2 = 0.5$	0.7828	0.5883	0.3842	0.0003	17.0982	-5.2751	39.4168	245
$\alpha_1 = 0.3, \alpha_2 = 0.7$	0.7827	0.4954	0.0038	0.0021	28.3687	-30.2577	53.1302	205



（a）The optimization process under $\alpha_1 = 0.7$ and $\alpha_2 = 0.3$



（b）The optimization process under $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$



（c）The optimization process under $\alpha_1 = 0.3$ and $\alpha_2 = 0.7$

Fig. 4 The optimization process under different weights in the objective function.

convergence effect has improved slightly, but the convergence speed is poor.

In the third experiment, set $C^R = [2\,1\,3\,4; 2\,8\,5\,5; 6\,8\,8\,8; 10\,10\,10\,10]$. The optimized QoS solutions under different user preferences are compared, and the results are shown in Table 5. It can be seen from Table 5 that the result of QoS optimization not only depends on the user's QoS preferences, but is closely related to the cost of renting resources. On one hand, the optimal QoS solution should be consistent with the user preference so as to promote the user's satisfaction; on the other hand, the optimal QoS solution should improve some QoS parameters with lower cost of renting resources to increase net profit of the service provider. Considering the aggregate utility of service provider and user, the service provider should give priority to improve the QoS parameters with lower cost according to the user's preferences.

In the fourth experiment, we compare the performance of the algorithm proposed in the paper (QoSUG) and the other two QoS optimization algorithms, including QoS optimization algorithm based on ant colony (QoSAC) and QoS optimization algorithm based on particle swarm (QoSPS). The former regards each QoS solution as a path through which ants can find food and use the objective function to guide the updating of pheromones. The latter regards each QoS solution as a particle in the population, uses the objective function to calculate fitness value of each particle, then follows the current optimal particle to search in the solution space. The experimental results are shown in Table 6. Specific parameters in the experiment are shown in Table 2.

As can be seen from Table 6, different QoS optimization algorithms have different optimization performance and convergence efficiency with the target of comprehensive utility

TABLE 4
The Effect of a on QoS Optimization Algorithm

A	SP utility after convergence	Convergence generation
1	36	421
2	52	64
2.1	53	89
2.2	53	93
2.3	55	121
2.4	57	140
2.5	61	118
3	62	350
4	66	3542

TABLE 5
The Results in the Third Experiment

weights	QoS solution after convergence				User preferences			
$\alpha_1 = 0.7$ $\alpha_2 = 0.3$	0.5707	0.2628	0.5376	0.9999	0.2	0.1	0.2	0.5
	0.5810	0.9996	0.0040	0.0017	0.05	0.85	0.05	0.05
	0.9989	0.2780	0.0087	0.9997	0.2	0.05	0.05	0.7
$\alpha_1 = 0.3$ $\alpha_2 = 0.7$	0.5353	0.0019	0.0018	0.9996	0.2	0.1	0.2	0.5
	0.0154	0.9969	0.0012	0.0029	0.05	0.85	0.05	0.05
	0.9233	0.0035	0.0001	0.9996	0.2	0.05	0.05	0.7
$\alpha_1 = 0.5$ $\alpha_2 = 0.5$	0.5431	0.2597	0.5268	0.9997	0.2	0.1	0.2	0.5
	0.0184	0.9988	0.0054	0.0024	0.05	0.85	0.05	0.05
	0.9987	0.0040	0.0009	0.9978	0.2	0.05	0.05	0.7

TABLE 6
The Results in the Fourth Experiment

Algorithm	Objective Function	Aggregate utility	Convergence generation
QoSUG	$\alpha_1 = 0.7\ \alpha_2 = 0.3$	8.7381	90
	$\alpha_1 = 0.3\ \alpha_2 = 0.7$	17.0982	245
	$\alpha_1 = 0.5\ \alpha_2 = 0.5$	28.3687	205
QoSAC	$\alpha_1 = 0.7\ \alpha_2 = 0.3$	8.0124	212
	$\alpha_1 = 0.3\ \alpha_2 = 0.7$	16.982	423
	$\alpha_1 = 0.5\ \alpha_2 = 0.5$	24.542	390
QoSPS	$\alpha_1 = 0.7\ \alpha_2 = 0.3$	5.234	61
	$\alpha_1 = 0.3\ \alpha_2 = 0.7$	10.778	180
	$\alpha_1 = 0.5\ \alpha_2 = 0.5$	17.812	166

of users and service providers. The QoS optimization method based on improved genetic algorithm (QoSUG) proposed in this paper is the best in terms of the accuracy and efficiency. Although QoSAC algorithm can also find a optimal QoS solution in the game, its convergence speed is significantly lower than that of QoSUG. QoSPS algorithm has a fast convergence, but its overall optimization performance is poor.

6.3 Summary

Through the above analysis, we obtained the following two conclusions:

1) When the provided QoS is completely consistent with the preferences of users, the game between users and service providers reaches an evolutionary equilibrium.
2) For service providers, the revenue of a service at the same price is affected by its resource cost and the user's preference. If a service provider wants to increase the revenue of the service while the cost of service remains unchanged, he/she should try his/her best to increase some QoS parameters with large weight of user preference and reduce some QoS parameters with high resource leasing cost as far as possible.

7 Conclusion

The problem of game based on utility between users and services providers in the information asymmetry cloud market is discussed in this paper. From the two aspects of theory study and algorithm design, the game model between users and service providers is analyzed and the QoS optimization algorithm based on the utility game is designed, and then the simulation experiments of the algorithm are performed. It provides a valuable guidance of theory and application for service providers to optimize the QoS. In future, we will further consider optimizing our model with more measured data from cloud platforms, and validate our model through actual effects of the service-based applications.

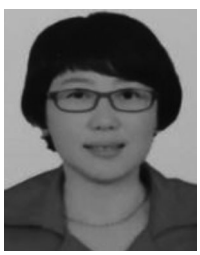
Acknowledgments

This work was supported in part by Natural Science Foundation of China under Grants 61662054, 61262082, Natural Science Foundation of Inner Mongolia under Grants 2019ZD15, 2019MS06029, Inner Mongolia Application Technology Research and Development Funding Project under Grant 201702168, Inner Mongolia Science and Technology Plan Project under Grant 2019GG372, Inner Mongolia Colleges and Universities Support Program for Young Scientific and Technological Talents under Grand NJYT-19-A02, Inner Mongolia Engineering Lab of Cloud Computing and Service Software, Inner Mongolia Key Laboratory of Data Processing and Social Computing, Inner Mongolia Engineering Lab of Big Data Analysis Technology.

References

[1] W. Novshek and H. Sonnenschein, "General equilibrium with free entry: A synthetic approach to the theory of perfect competition," *J. Econ. Literature*, vol. 25, no. 3, pp. 1281–1306, Mar. 1987.
[2] S. H. H. Madni, M. S. A. Latiff, and Y. Coulibaly, "Resource scheduling for infrastructure as a service (IaaS) in cloud computing," *J. Netw. Comput. Appl.*, vol. 68, no. 6, pp. 173–200, Jun. 2016.
[3] X. Lin and C. Q. Wu, "On scientific workflow scheduling in clouds under budget constraint," in *Proc. IEEE 42nd Int. Conf. Parallel Process.*, 2013, pp. 90–99 .
[4] H. Arabnejad and J. G. Barbosa, "A budget constrained scheduling algo-rithm for workflow applications," *Grid Comput.*, vol. 12, no. 4, pp. 665–679, Dec. 2014.
[5] H. Liu, D. Xu, and H. Miao, "Ant colony optimization based service flow scheduling with various QoS requirements in cloud computing," in *Proc. IEEE 1st Acis Int. Symp. Softw. Netw. Eng.*, 2011, pp. 53–58.
[6] T. Chen, R. Bahsoon, and G. Theodoropoulos, "Dynamic QoS optimization architecture for Cloud-based DDDAS," *Procedia Comput. Sci.*, vol. 18, no. 1, pp. 1881–1890, Jan. 2013.

[7] T. S. Li and X. X. Zhang, "Differentiated service-based evolutionary game scheduling algorithm for cloud computing," *J. Beijing Univ. Posts Telecommun.*, vol. 36, no. 1, pp. 41–45, Jan. 2013.
[8] M. Zhu *et al.*, "An approach for QoS-aware service composition with graphplan and fuzzy logic," *Procedia Comput. Sci.*, vol. 141, pp. 56–63, 2018.
[9] H. Ma *et al.*, "Collaborative optimization of service composition for data-intensive applications in a hybrid cloud," *IEEE Trans. Parallel Distrib.*, vol. 30, no. 5, pp. 1022–1035, May 2019.
[10] S. Chaisiri, B. S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Trans. Serv. Comput.*, vol. 5, no. 2, pp. 164–177, Second Quarter 2012.
[11] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, "Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds," *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 158–169, Jan. 2013.
[12] Z. Wu, X. Liu, and Z. Ni, "A market-oriented hierarchical scheduling strategy in cloud workflow systems," *J. Supercomput.*, vol. 63, no. 1, pp. 256–293, Jan. 2013.
[13] Y. Sun, J. White, and S. Eade, "ROAR: A QoS-oriented modeling framework for automated cloud resource allocation and optimization," *J. Syst. Softw.*, vol. 116, pp. 146–161, Oct. 2016.
[14] Y. C. Lee, C. Wang, and A. Y. Zomaya, "Profit-driven scheduling for cloud services with data access awareness," *J. Paraller Distrb. Comput.*, vol. 72, no. 4, pp. 591–602, Apr. 2012.
[15] H. N. Hoang, S. L. Van, and N. M. Han, "Admission control and Scheduling algorithms based on ACO and PSO heuristic for optimizing cost in cloud computing," in *Recent Developments in Intelligent Information and Database Systems*, Berlin, Germany: Springer, Feb. 2016, pp. 15–28.
[16] L. Wu, S. K. Garg, and R. Buyya, "SLA-Based resource allocation for software as a service provider (SaaS) in cloud computing environments," in *Proc. IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, 2011, pp. 195–204.
[17] S. Deng *et al.*, "Cost performance driven service mashup: A developer perspective," *IEEE Trans. Parall. Distrib.*, vol. 27, no. 8, pp. 2234–2247, Aug. 2016.
[18] M. M. Rahman, C. Despins, and S. Affes, "Design optimization of wireless access virtualization based on cost & QoS Trade-off utility maximization," *IEEE Commun.*, vol. 15, no. 9, pp. 6146–6162, Sep. 2016.
[19] L. Jiao, J. Li, and T. Y. Xu, "Cost optimization for online social networks on geo-distributed clouds," *IEEE Trans. Netw.*, vol. 24, no. 1, pp. 99–112, Feb. 2016.
[20] T. Chen and R. Bahsoon, "Self-adaptive trade-off decision making for au-toscaling cloud-based services," *IEEE Trans. Services Comput.*, vol. 10, no. 4, pp. 618–632, Jul./Aug. 2017.
[21] Y. Huo *et al.*, "Multi-objective service composition model based on cost-effective optimization," *Appl. Intell.*, vol. 2017, no. 4, pp. 1–19, 2017.
[22] Z. Zhu, G. Zhang, and M. Li, "Evolutionary multi-objective workflow scheduling in cloud," *IEEE Trans. Parallel Distrib.*, vol. 27, no. 5, pp. 1344–1357, May. 2016.
[23] M. Chen *et al.*, "Statistical model checking-based evaluation and optimization for cloud workflow resource allocation," *IEEE Trans. Cloud Comput.*, vol. 8, no. 2, pp. 443–458, Third Quarter 2020.
[24] J. Zhou *et al.*, "An adaptive multi-population differential artificial bee colony algorithm for many-objective service composition in cloud manufacturing," *Inf. Sci.*, vol. 456, pp. 50–82, 2018.
[25] Z. Y. Zhang, L. Cherkasova, and B. T. Loo, "Optimizing cost and perfor-mance trade-offs for mapreduce job processing in the cloud," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, 2014, pp. 1–8.
[26] A. Idrissi and F. Zegrari, "A new approach for a better load balancing and a better distribution of resources in cloud computing," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 10, pp. 1165–1170, 2015.
[27] S. Singh *et al.*, "SOCCER: Self-optimization of energy-efficient cloud resources," *Cluster Comput.*, vol. 19, no. 4, pp. 1787–1800, 2016.
[28] S. Singh and I. Chana, "QRSF: QoS-aware resource scheduling framework in cloud computing," *J. Supercomput.*, vol. 71, no. 1, pp. 241–292, 2015.
[29] S. Xue *et al.*, "QET: A QoS-based energy-aware task scheduling method in cloud environment," *Cluster Comput.*, vol. 20, no. 1, pp. 1–14, 2017.
[30] J. L. L. Simarro, R. Moreno-Vozmediano, and R. S. Montero, "Dynamic placement of virtual machines for cost optimization in multi-cloud envi-ronments," in *Proc. IEEE Int. Conf. High Perform. Comput. Simul.*, 2011, pp. 1–7.
[31] H. Xu and B. Li, "Maximizing revenue with dynamic cloud pricing: The infinite horizon case," in *Proc. IEEE Int. Conf. Commun.*, 2012, pp. 2929–2933.
[32] D. Paul, W. D. Zhong, and S. K. Bose, "Energy efficient cloud service pricing," *J. Netw. Comput. Appl.*, vol. 64, no. C, pp. 98–112, Mar. 2016.
[33] P. Debdeep, Z. Wen-De, and B Sanjay, "Energy aware pricing in a three-tiered cloud service market," *Electronics*, vol. 65, no. 5, pp. 1–19, 2016.
[34] C. Lee, P. Wang, and D. Niyato, "A Real-time group auction system for Effi-cient allocation of cloud internet applications," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 251–268, Mar./Apr. 2015.
[35] A. S. Alrawahi, K. Lee, and A. Lotfi, "A multiobjective QoS model for trading cloud of things resources," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9447–9463, Dec. 2019.
[36] Y. Wang, J. T. Zhou, and H. Y. Tan, "CC-PSM: A preference-aware selection model for cloud service based on consumer community," *Math. Problem Eng.*, vol. 2015, no. 7, pp. 1–13, Oct. 2015.
[37] Y. Wang *et al.*, "Research on QoS optimization method of cloud service based on utility game between users and service providers in the cloud market," in *Proc. IEEE Int. Conf. Services Comput.*, 2017, pp. 297–304.
[38] R. Gibbons, *A Primer in Game Theory*. New York, NY, USA: Pearson Academic, 1992, pp. 33–36 .
[39] X. Huang, "UsageQoS: Estimating the QoS of web services through online user communities," *ACM Trans. Web.*, vol. 8, no. 1, pp. 1–31, 2013.
[40] J. M. Smith, "Evolution and theory of games," *Amer. Scientist*, vol. 64, no. 1, pp. 41–45, Jan. 1982.
[41] J. Hirshleifer, "Evolutionary models in economics and law: Cooperative versus conflict strategies," *Res. Law Econ.*, vol. 1982, no. 4, pp. 1–60, Apr. 1982.
[42] Y. P. Chen, T. L. Yu, K. Sastry, and D. E. Goldberg, "A survey of linkage learning techniques in genetic and evolutionary algorithms," IlliGAL, Univ. Illinois Urbana-Champaign, Illinois Genetic Algorithms Lab., Rep. no. 2007014, pp. 2–25, Apr. 2007.



Yan Wang received the PhD degree in computer science from Inner Mongolia University, in 2015. She is currently lecturer and a BS supervisor with the College of Computer Science, Inner Mongolia University University. Her research interests include service computing, formal methods and software technology.

Jian-Tao Zhou received the PhD degree from Tsinghua University, in 2005. She is currently a professor and a PhD supervisor in College of Computer Science, Inner Mongolia University. Her current research interests concentrate on network computing and formal methods.

Xiaoyu Song received the PhD degree from the University of Pisa, Italy, in 1991. From 1992 to 1998, he was on the faculty with the University of Montreal, Canada. He joined the Department of Electrical and Computer Engineering, Portland State University, in 1998, where he is currently a professor. He was an editor of the *IEEE Transactions on VLSI Systems* and *IEEE Transactions on Circuits and Systems*. He was awarded an Intel Faculty Fellowship from 2000 to 2005. His research interests include formal methods, design automation, embedded systems and emerging technologies.