



QoS-Aware Cloud Service Optimization Algorithm in Cloud Manufacturing Environment

Wenlong Ma^{1,2,*}, Youhong Xu¹, Jianwei Zheng² and Sadaqat ur Rehman³

¹School of Information Engineering, Quzhou College of Technology, Quzhou, 324000, China

²School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, 310023, China

³Department of Natural and Computing Science, University of Aberdeen, Scotland, Aberdeen, AB243FX, UK

*Corresponding Author: Wenlong Ma. Email: qzmwlmwl@126.com

Received: 27 March 2022; Accepted: 24 June 2022; Published: 23 June 2023

Abstract: In a cloud manufacturing environment with abundant functionally equivalent cloud services, users naturally desire the highest-quality service(s). Thus, a comprehensive measurement of quality of service (QoS) is needed. Optimizing the plethora of cloud services has thus become a top priority. Cloud service optimization is negatively affected by untrusted QoS data, which are inevitably provided by some users. To resolve these problems, this paper proposes a QoS-aware cloud service optimization model and establishes QoS-information awareness and quantification mechanisms. Untrusted data are assessed by an information correction method. The weights discovered by the variable precision Rough Set, which mined the evaluation indicators from historical data, providing a comprehensive performance ranking of service quality. The manufacturing cloud service optimization algorithm thus provides a quantitative reference for service selection. In experimental simulations, this method recommended the optimal services that met users' needs, and effectively reduced the impact of dishonest users on the selection results.

Keywords: Cloud manufacturing; quality of service; optimization algorithm; rough set

1 Introduction

As a new type of networked manufacturing, cloud manufacturing utilizes network and cloud manufacturing service platforms to organize online manufacturing resources (manufacturing clouds) that meet user needs, thus providing users with various types of on-demand manufacturing services [1]. Therefore, the manufacturing cloud service deployed on the network contains an abundant amount of information. Many manufacturing cloud services have the same or similar functions, but vary in their service quality. Users naturally want to select the service(s) offering the highest quality of service (QoS) [2–5]. The optimization of QoS-based services must solve two simultaneous problems: 1) describing, quantifying, and monitoring the QoS attributes of manufacturing cloud services, 2) efficiently selecting the optimal service that meets the QoS requirements of the user among a number of similar services. The first problem can be approached through real-time sampling, detection, user feedback, or historical data



obtained through data mining [6–9]; the second problem requires comprehensive consideration of various QoS performance indicators of the service, and evaluation of their importance. Finally, a flexible multi-indicator decision is made following a certain strategy, and the final solution is returned after ranking and selecting the decisions [10–12].

In fact, the QoS experience reported by users is sometimes inconsistent with the QoS information provided by the manufacturing cloud service registry. In some cases, this inconsistency is significant [13–16]. On the one hand, some service providers driven by self-interest may deliberately exaggerate their QoS. On the other hand, not all users leave honest evaluations of the service execution results. Some users deliberately talk-up or talk-down the QoS, leaving good or bad reviews about the service provider's honesty with malicious intent. These situations mainly occur because the QoS value provided by service registries lacks the corresponding honesty measurements and an information correction mechanism.

In the present paper, the abovementioned problems are resolved by a QoS-aware cloud service optimization model that meets the specific needs of a manufacturing industry in the cloud-manufacturing environment. First, the collection, quantification, and information correction methods of QoS evaluation indicators suitable for cloud-manufacturing businesses are discussed. Next, the weights of various evaluation indicators are obtained from a large numbers of historical data records through a variable precision rough set. These weights are combined with the preset evaluation indicator weights of users with specific business needs and personal preferences. Finally, multi-indicator decisions are made by mining the weights of the indicators, thereby finding the optimal service to recommend to users.

The main contributions of this paper are summarized as follows:

- 1) A novel QoS information model of manufacturing cloud services accommodating the characteristics of the manufacturing industry is proposed.
- 2) A QoS information correction method that effectively reduces the influence of untrusted data on the selection results is proposed.
- 3) Theoretical and experimental analyses confirm the effectiveness of the proposed optimization algorithm for manufacturing cloud services, and the objectiveness of the proposed variable precision rough set weight method.

The rest of this paper is organized as follows. In Section 2, we give an overview of relevant work. Section 3 illustrates QoS-information awareness and quantification mechanisms. A QoS-aware cloud service optimization model are given in Section 4. In Section 5 we present the main implementation and the experiment to verify the efficiency of our method. This is followed by Conclusions and Future Work in Section 6.

2 Relevant Work

Optimization of manufacturing cloud services has followed two main research directions: a. finding appropriate resources that match the functional description, b. finding appropriate resources based on non-functional descriptions. In the first approach, the search is completed when the detailed functional description of the manufacturing resource matches the required resource functions. This approach is suitable when initially searching the service. The second approach intends to satisfy the functional requirements of the service to determine the best service based on certain non-functional requirements. The best manufacturing resources are found by matching the service quality and resource requirements of many similar services. In a non-functional resource search, Zhou et al. [17] proposed that both the resource service management QoS and network performance QoS should be comprehensively considered in the manufacturing grid QoS evaluation. They also proposed a QoS evaluation model with 11 indicators, including time and price. Liu et al. [18] proposed an extensible calculation model based on

general QoS and special QoS attributes. Yau et al. [19] proposed a QoS sorting algorithm based on user satisfaction alone, without considering the weights of the QoS attributes. Tao et al. [20] proposed a non-functional QoS evaluation method and a resource service optimization algorithm for manufacturing-resource services. Their method constructs an intuitionistic fuzzy set, combining intuitionistic fuzzy set theory and its corresponding operation rules.

The reliability of QoS data has attracted many researchers of Web services. Xu et al. [21] measured the reliability of each user in a cloud service-selection framework based on user reputation, and proposed a novel calculation method of user reputation. Xiong [22] considered the impact of similarity among users, and proposed a peer trust model that evaluates the reliability of users. Wang et al. [23] assessed service reputation by feedback verification, confirmation, and feedback tests. A trust framework that enables services to build reliable trust relationships has also been proposed [24,25].

As indicated above, QoS-aware selection of Web services has been studied from many perspectives, providing valuable references for selecting manufacturing cloud services. However, unlike Web services, manufacturing cloud services must abide by certain special requirements of the industry. The research object of Web services is developing computing and software resources. Manufacturing cloud services must provide not only the computing and software resources, but also other manufacturing resources and manufacturing capabilities. To cope with the large number of complex manufacturing services, the description model of the QoS information should differ from that of Web services. Using a non-functional description method, the present model attempts to improve the effectiveness of optimization models for cloud manufacturing services.

3 QoS Awareness and Information Correction of Manufacturing Cloud Services

3.1 QoS-aware Model of Manufacturing Cloud Services

In this paper, manufacturing cloud services are divided into hardware and software cloud services. Hardware cloud services produce and manufacture the equipment, whereas software cloud services deliver the software resources. To develop a QoS-aware model for manufacturing cloud services, we first define the information of the manufacturing cloud service.

Definition 1. The QoS information of a manufacturing cloud service S can be modeled as $S_{QoS} = \{p_{price}, p_{time}, p_{rel}, p_{ava}, p_{hon}\}$, where p_{price} is the price of the cloud service. If S is a hardware cloud service, the price includes the outsourcing and transportation costs. Denoting the current average outsourcing and transporting costs of a single product by C_r and C_t respectively, we have $p_{price} = C_r + C_t$; if S is a software cloud service, then p_{price} is the fee of using service S .

p_{time} represents the time interval between the calling of service S and the receiving of a response. This value indicates the responsiveness of the cloud service to a user's request. If S is a hardware cloud service, the processing and transportation times of the task product are T_{proc} and T_{trans} respectively, and $p_{time} = T_{com} + T_{delay}$; if S is a software cloud service requiring computational time T_{com} , the time delay between issuing the call command and the start of the S execution is T_{delay} , so we have $p_{time} = T_{com} + T_{delay}$. The above time and price samplings and measurement methods obviously differ from those of traditional services; in particular, they fully consider the business background of the cloud service as a product manufacturing service.

The reliability of the service (p_{rel}) defines the ability of the cloud service to $\sum service(p_{ava})$ is the probability of normal service operation, and the honesty of the service (p_{hon}) defines the extent to which the user complies with the agreement after the service is fulfilled. The latter is determined by average evaluation. Suppose that service S is called K times during time interval (t_1, t_2) . Let the number of normal

responses be K_n , the time of no failures be t_n , and the users' evaluations at each time be E_i . The above three indices are then defined as

$$p_{rel} = K_n/K \quad (1)$$

$$p_{ava} = t_n/(t_2 - t_1); \quad (2)$$

$$p_{hon} = \frac{1}{K_n} \sum_{i=1}^{K_n} E_i. \quad (3)$$

3.2 QoS Information Correction

In the proposed method, dishonest evaluations are detected by a monitor placed in the cloud manufacturing service platform, which collects, checks, and verifies the feedback data of users. Data which are too large or too small, which likely represent the malicious attack data, are filtered out. Whether data represent a malicious attack is judged by user collaborative filtering. Evaluations given by the same users using the same services are expected to vary only slightly.

Suppose that user X and user Y access the same service $S = \{s_1, s_2, \dots, s_k\}$. The honest evaluations of the k services given by users X and Y can be expressed as vector spaces $U_x: \{(s_1, x_1), (s_2, x_2), (s_3, x_3), \dots, (s_k, x_k)\}$ and $U_y: \{(s_1, y_1), (s_2, y_2), (s_3, y_3), \dots, (s_k, y_k)\}$, respectively. Usually, the degree of similarity between two vectors is given by the Euclidean Distance between the vectors. Defining the Euclidean Distance between the vector spaces U_x and U_y as $d(U_x, U_y)$, the distance between users X and Y is given by:

$$d(U_x, U_y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (4)$$

The above method gives the set of users $U = \frac{1}{m} \sum_{i=1}^m h_i \{u_1, u_2, \dots, u_m\}$ similar to user X . Suppose that m "X-type" users access a service. If user A evaluates the honesty of a new service s as h_x , and the evaluation given by the i -th user is h_i , then the average evaluation is $ha = \frac{1}{m} \sum_{i=1}^m h_i$ and the honesty of this user's evaluation is computed as:

$$Hon = \begin{cases} 0, & |h_x - ha|/ha > 1 \\ 1 - \frac{|h_x - ha|}{ha}, & |h_x - ha|/ha \leq 1 \end{cases} \quad (5)$$

The final evaluation of the users, given by $h_x \times Hon$, is then written into the QoS database by the monitor. Therefore, the setting of the user evaluation honesty can adjust or weaken the subjective factors of the users, and filter the outliers that might signify malicious attack data.

3.3 Collection and Quantification of QoS Information

The comprehensive QoS indicators of manufacturing cloud services are derived from the historical records of each evaluation indicator. The collection sources can be divided into two categories: QoS indicators of the network performance of the software cloud services (such as service computing time and network transmission delay), and QoS indicators of the manufacturing resources of hardware cloud services. The former can be extracted from the QoS database of the corresponding cloud manufacturing service platform, and the latter are generally provided by users (service providers, service users and

platform operators) of the cloud manufacturing platform. The price of outsourcing, transportation, and other services can be directly obtained from the user input values on the manufacturing service platform.

Let $S = \{s_1, s_2, \dots, s_m\}$ be a set of candidate services providing similar services in the manufacturing cloud pool. The QoS attribute value of each candidate service is inserted into a vector $p_i = \{p_{i1}, p_{i2}, \dots, p_{in}\}$. The QoS information of the relevant candidate services when selecting cloud manufacturing services is then expressed as:

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_m \end{bmatrix} = \begin{bmatrix} p_{11}, p_{12}, \dots, p_{1n} \\ p_{21}, p_{22}, \dots, p_{2n} \\ \dots, \dots, \dots, \dots \\ p_{m1}, p_{m2}, \dots, p_{mn} \end{bmatrix}. \tag{6}$$

To comprehensively evaluate a cloud service, we must quantify the individual QoS indicators. As different QoS attributes have very different values and measurement units, they cannot be directly calculated. Instead, their values must be normalized to facilitate the multi-objective decision-making. This paper uses the following unified quantitative utility function [26]:

$$P_{ij} = \begin{cases} \frac{p_j^{\max} - p_{ij}}{p_j^{\max} - p_j^{\min}}, & p_j^{\max} - p_j^{\min} \neq 0 \\ 1, & p_j^{\max} - p_j^{\min} = 0 \end{cases}, \tag{7}$$

$$P'_{ij} = \begin{cases} \frac{p_{ij} - p_j^{\min}}{p_j^{\max} - p_j^{\min}}, & p_j^{\max} - p_j^{\min} \neq 0 \\ 1, & p_j^{\max} - p_j^{\min} = 0 \end{cases}, \tag{8}$$

where p_j^{\max} and p_j^{\min} represent the maximum and minimum values of the j -th attribute among all candidate services in service class S , respectively. The QoS attributes of each candidate service can be divided into forward and reverse attributes. Forward attribute implies that a larger value corresponds to better service quality, such as reliability, availability, and service honesty. This can be calculated using Eq. (8). Reverse attribute implies that a smaller value corresponds to better service quality, such as price and service time. This can be calculated using Eq. (7). After applying the unified quantization to Eq. (6), the following matrix is obtained:

$$P' = \begin{bmatrix} P_1 \\ P_2 \\ \dots \\ P_m \end{bmatrix} = \begin{bmatrix} P_{11}, P_{12}, \dots, P_{1n} \\ P_{21}, P_{22}, \dots, P_{2n} \\ \dots, \dots, \dots, \dots \\ P_{m1}, P_{m2}, \dots, P_{mn} \end{bmatrix} \tag{9}$$

4 QoS-Based Manufacturing Cloud Service Optimization Model

To optimize a manufacturing cloud service, the performance of each QoS indicator of the service must be comprehensively considered, and the indicators must be weighted by their importance. This problem constitutes a typical multi-objective decision problem. Commonly, the weight ratio is calculated between the decision factors in the decision analysis and the weight calculation. The weights are calculated by the analytic hierarchy process or a similar method [27,28]. As these methods input the values of artificial experiences, their decision outputs are strongly subjective. To improve the objectivity, this paper calculates the decision factors by rough set (Rough Set, RS) theory. The RS is weighted by the importance of the attributes of each decision condition, which can be analyzed directly from the historical data records.

4.1 Attribute Weight Calculation of RS with Variable Precision

Before applying RS theory, we must cluster the QoS attribute values of each manufacturing cloud service. This paper adopts the k -center clustering method [29]. To reduce the interferences from the inevitable noises in actual manufacturing businesses, the RSs were processed by Ziarko's rough-set model with variable precision [30]. This method relaxes the approximate boundaries of standard RSs, extending their upper and lower values to a defined precision level $\beta \in [0, 0.5)$. When $\beta = 0$, the rough-set model with variable precision reduces to the classical rough-set model.

Let the tetrad $I = (U, a = C \cup D, V, F)$ be the QoS information decision-making system, and U be the non-empty subset of the finite domain of the instance object, where $U = \{x_1, x_2, \dots, x_m\}$ represents the m history records. Also let A be a non-empty finite set of QoS attributes, $C = \{a_1, a_2, \dots, a_n-1\}$ be the condition attribute set of QoS, $D = \{a_n\}$ be the decision attribute set, V be the value domain of the attribute set, and F be the information function of each attribute mapping to the value domain.

Definition 2. Let X and Y be non-empty subsets in a finite domain. If $X \subseteq Y$ where \subseteq is a partially ordered relationship, then

$$c(X, Y) = \begin{cases} 1 - \frac{|X \cap Y|}{|X|}, & X \neq \phi \\ 0, & X = \phi \end{cases} \quad (10)$$

where $|X|$ is the cardinality of the set, defining the number of objects in the equivalent class. $c(X, Y)$ represents the relative error classification rate of X with respect to Y . If $0 \leq \beta < 0.5$, then there exists a majority inclusion relationship $Y \stackrel{\beta}{\supseteq} X \Leftrightarrow c(X, Y) \leq \beta$. When $c(X, Y) = 0$, there exists a standard inclusion relationship from Y to X .

In the information decision-making system of QoS, this paper sets $X = U/C = \{X_1, X_2, \dots, X_{|U/C|}\}$, defining the equivalent class divided by the domain U according to the conditional attribute set C . Meanwhile, $Y = U/D = \{Y_1, Y_2, \dots, Y_{|U/D|}\}$ is the equivalent class divided by the domain U according to the decision attribute set D .

Definition 3. Given a $\beta \in [0, 0.5)$, the lower approximation β of Y_j with respect to the conditional attribute set C is given by:

$$Pos_{X_k}^{\beta} = \cup \{Y \in U/D \mid c(X_i, X_j) \leq \beta\} \quad (11)$$

The positive region $Pos_{X_k}^{\beta}(Y_j)$ of β , also denoted as $C_{\beta}(Y_j)$, is the set of lower approximations $\{C_{\beta}(Y_1), C_{\beta}(Y_2), \dots, C_{\beta}(Y_{|U/D|})\}$ of all decision classes Y_j , representing the probability distribution of U/C in each decision class.

Definition 4. The information amount of a conditional attribute [30] reflects the ability of the attribute to classify data objects. The greater the amount of information, the stronger is the ability to classify objects. The information amount of conditional attribute α_k , is calculated as:

$$\gamma(\alpha_k) = \frac{1}{|U|^2} \sum_{i=1}^{|U/\alpha_k|} |X_i|^2, \alpha_k \in C \quad (12)$$

where X_i is an equivalence class of the conditional attribute α_k in U , and $|X_i|$ is the cardinality of that equivalence class.

Definition 5. The dependency degree [30] of a conditional attribute reflects the dependency degree of the decision-attribute classification of that attribute. The greater the dependency degree, the more critical is the attribute. In this paper, the dependency degree of conditional attribute α_k is expressed as:

$$\lambda(\alpha_k) = \frac{\sum_{j=1}^{|U/D|} |Pos_{\alpha_k}^{\beta}(Y_j)|}{|U|} \quad (13)$$

The information amount and dependency of conditional attributes represent different aspects of the attribute importance, and must be comprehensively evaluated. In this paper, both aspects are considered equally important. The dependence and information amount of each attribute, calculated by Eqs. (12) and (13) respectively, provide scattered information. This information must then be normalized to give the importance (weight) of the conditional attribute. This article proposes the following normalization method:

$$w_k = \frac{(\gamma(a_k) + \lambda(a_k))/2}{\sum_{k=1}^{n-1} (\gamma(a_k) + \lambda(a_k))/2}, \text{ with, } \sum_{k=1}^{n-1} w_k = 1 \quad (14)$$

4.2 Optimization Algorithm of the Manufacturing Cloud Service

Let $Z = \{z_1, z_2, \dots, z_n\}$ be the user-assigned weight of each QoS attribute, and $W = \{w_1, w_2, \dots, w_n\}$ be the weight calculated by the RS. $S = \{s_1, s_2, \dots, s_m\}$ is known as the candidate set of entities providing similar services in the manufacturing cloud pool. The QoS attribute value of each candidate service is compiled into the vector $P_s = \{p_1, p_2, \dots, p_n\}$. The main steps of the proposed optimization algorithm for manufacturing cloud services are listed below.

Algorithm 1: Optimization algorithm for manufacturing cloud services.

Inputs: 1) QoS decision information system $I = (U, A = C \cup D, V, F)$, where U is the domain, C is the conditional attribute set, and D is the decision attribute set; 2) Weight vector Z given by the user.

Output: Comprehensive performance ranking sequence of QoS.

Step 1 Formulate the matrix P of QoS attribute awareness values in the decision information system I using Eqs. (1)–(3), (6).

Step 2. Normalize the matrix P to P' by Eqs. (7) and (8).

Step 3. Select the maximum value in each column of P' matrix, and compile it into the QoS vector $P^{\max} = \{p_1^{\max}, p_2^{\max}, \dots, p_n^{\max}\}$ of the ideal service. Calculate the product of P^{\max} and the user-provided weight vector Z .

Step 4: Calculate the information amount of each conditional attribute by Eq. (12), the dependency degree of each conditional attribute by Eq. (13), and the weight vector W of each attribute by Eq. (14).

Step 5: Compute the weighted product of the i -th row vector P_i and the weight vector W in the P' matrix, and calculate the distance between P_i and the vector P^{\max} , that is, the similarity $Sim(S_i) =$

$$1 - \sqrt{\sum_{j=1}^{n-1} (p_j^{\max} * Z_j - P_{ij} * w_j)^2} \text{ between service } S_i \text{ and the ideal service.}$$

Step 6: Output the service sequence in descending order of Sim values.

5 Experimental Procedure and Results

5.1 Experimental Design

Assisted by the telecom cloud computing platform, a prototype of the cloud manufacturing service test platform was designed using jdk15+MyEclipse2019.4.0 as the Integrated Drive Electronics development environment, Tomcat 8.5 as the server, and MySQL 8.0 and Sybase16.0 as the database and its design tools, respectively. The call of computer nodes to cloud manufacturing service platform services was simulated in Mpiblast, a common distributed application software. Because the description and definition of manufacturing cloud services are highly autonomous and diverse, no common service benchmark database has been recognized by the vast majority of scholars. Therefore, a standard test set is currently lacking. Most service-selection tests of manufacturing cloud services instead use randomly generated test data. The cloud manufacturing service test platform established in this paper includes 1200 manufacturing cloud services for testing.

As an example, we consider production by a door handle manufacturer. The final production process is electroplating the door handle. To meet the production requirements, 2,0000 door handles must be electroplated within 20 days. The manufacturer publishes these requirements to the cloud manufacturing service platform. The QoS attribute value of each service is extracted from the service platform QoS database. After receiving a service request, the platform-related module queries the hardware cloud service resource library in the manufacturing cloud pool. Suppose that five similar hardware cloud services in the pool can meet the functional requirements of the manufacturer. The manufacturer bases its decision on the cost, time, reliability, availability, and honesty indicators of the five services. The services and their QoS indicator values are listed in [Table 1](#).

Table 1: Similar services that meet the functional requirements of a user

Service	Cost (yuan/piece)	Time (days)	Reliability	Availability	Honesty
S1	0.60	14	30/30	0.51	4.21
S2	0.61	17	40/42	0.82	4.72
S3	0.64	13	2/10	0.61	1.52
S4	0.65	17	3/3	0.86	4.92
S5	0.63	14	5/5	0.91	4.42

The cost in [Table 1](#) is obtained by summing the processing costs and per-kilometer transportation costs of a single product. The time is the number of days required for processing and transporting the product. Service 2 has a reliability value of 40/42, indicating that this service was called 42 times from the QoS database, and provided 40 normal responses. In a practical scenario, the historical record of Service S2 revealed 42 bids for its service, and 40 completed contracts among those bids. The availability describes the probability of a service's failure-free time during the effective period of the database record. Honesty (range [0, 5]) is the average evaluation obtained after completing all bid-winning projects. The honesty data are corrected by the QoS monitor in the cloud manufacturing service platform (see [Eq. \(5\)](#)). The cost and time indicators are normalized by [Eq. \(2\)](#), and the other indicators are processed by [Eq. \(3\)](#). The normalized data are listed in [Table 2](#).

Table 2: Normalized QoS indicators of the similar services listed in Table 1

Service	Cost (yuan/piece)	Time (days)	Reliability	Availability	Honesty
S ₁	1.01	0.75	1.00	0.00	0.78
S ₂	0.81	0.00	0.96	0.75	0.95
S ₃	0.21	1.00	0.00	0.25	0.00
S ₄	0.00	0.00	1.00	0.92	1.00
S ₅	0.42	0.75	1.00	1.00	0.86

Based on the attribute weight calculation of the RS, the weight ratio of each attribute was obtained by analyzing the historical records. The presented experiment accessed the historical records of 5116 manufacturing resources in the manufacturing cloud pool. To demonstrate the reasoning process, we randomly selected eight records as the reasoning sample. The cost and time factors likely differ among the processed products, and the data are not directly comparable. The measurement parameter was the ratio of the bid quote in the record to the bid price of the current business, and the time factor was analyzed in terms of the advance time ratio. For example, service S1 in Table 1 participated in the bidding three days in advance of its manufacturing time (10 d), so its time factor was 0.3. The bidding history of the eight randomly selected manufacturing-resource records are listed in Table 3.

Table 3: Bid histories of the randomly selected manufacturing resources

Record	Cost ratio	Advance time ratio	Reliability	Availability	Honesty	Bid won
x ₁	0.97	0.91	0.98	0.91	0.98	Yes
x ₂	0.91	0.72	0.83	0.88	0.88	Yes
x ₃	0.71	0.11	0.51	0.78	0.74	No
x ₄	0.86	0.52	0.75	0.89	0.81	Yes
x ₅	0.76	0.42	0.66	0.76	0.72	No
x ₆	0.79	0.52	0.73	0.78	0.87	No
x ₇	0.66	0.11	0.82	0.88	0.77	No
x ₈	0.86	0.22	0.69	0.89	0.96	Yes

For processing by the RS (which can process only clustered data), the data in the historical database were clustered by the *k-center* clustering method described in [29]. The number of clusters was set to 5, and the clustered data are shown in Table 4.

According to the data in Table 4, the bid-winning decision-making class was $Y_1 = U/d_1 = \{x_1, x_2, x_4, x_8\}$, and the bid-losing decision-making class was $Y_2 = U/d_2 = \{x_3, x_5, x_6, x_7\}$. For conditional attribute α_5 , the records were divided into equivalent classes $U/\alpha_5 = \{\{x_1, x_8\}\{x_2, x_4, x_6\}, \{x_3, x_5, x_7\}\}$, and the information amount of this conditional attribute was calculated by Eq. (12): $\gamma(\alpha_5) = 44/128$. The information amounts of the other attributes were similarly obtained as $\gamma(\alpha_1) = 64/128$, $\gamma(\alpha_2) = 36/128$, $\gamma(\alpha_3) = 28/128$, and $\gamma(\alpha_4) = 68/128$.

Table 4: Information decision-making in the bid histories of the manufacturing services

Record	Cost ratio (α_1)	Advance time ratio (α_2)	Reliability (α_3)	Availability (α_4)	Honesty (α_5)	Bid won (D)
x ₁	1	1	1	1	1	1
x ₂	1	2	2	1	2	1
x ₃	2	4	5	2	3	0
x ₄	1	3	3	1	2	1
x ₅	2	3	4	2	3	0
x ₆	2	3	3	2	2	0
x ₇	2	4	2	1	3	0
x ₈	1	4	4	1	1	1

Setting $\beta = 0.4$ in Eq. (11), the β -lower approximate distributions of Y_1 and Y_2 with respect to conditional attribute α_5 were calculated as $Pos_{X_5}^{\beta}(Y_1) = \{x_1, x_8\}$ and $Pos_{X_5}^{\beta}(Y_2) = \{x_3, x_5, x_7\}$, respectively. Eq. (13) gives the dependency degree of conditional attribute α_5 as $\lambda(\alpha_5) = 10/16$. Similarly, we obtained $\lambda(\alpha_1) = 16/16$, $\lambda(\alpha_2) = 6/16$, $\lambda(\alpha_3) = 4/16$, $\lambda(\alpha_4) = 14/16$. By Eq. (11), we then obtained $w_1 = 0.31$, $w_2 = 0.15$, $w_3 = 0.10$, $w_4 = 0.29$, and $w_5 = 0.15$. After analyzing a large number of collected experimental data and considering the suggestions of manufacturing experts, the pre-entered attribute preference weights were set to $Z = \{0.4, 0.1, 0.1, 0.2, 0.2\}$. Using the data information in Table 2, the similarities of the services were calculated by Algorithm 1, thus obtaining $sim(S_1) = 0.78$, $sim(S_2) = 0.82$, $sim(S_3) = 0.59$, $sim(S_4) = 0.60$, and $sim(S_5) = 0.73$. After ranking the performances of the services as $S_2 > S_1 > S_5 > S_4 > S_3$, service s_2 was selected as the optimal choice.

If the weights of the QoS indicators are ignored or set to the same value, the data of Table 1 yield $sim(S_1) = 0.73$, $sim(S_2) = 0.63$, $sim(S_3) = 0.45$, $sim(S_4) = 0.55$, $sim(S_5) = 0.76$, and the services are ranked as $S_5 > S_1 > S_2 > S_4 > S_3$. Service S_5 receives the best performance score because it requires a shorter time and has a high contract completion rate, but its cost is high and its honesty is only average. Overall, S_5 gives a balanced performance. When the weights are the weights Z preset by the user, the data of Table 1 give $sim(S_1) = 0.76$, $sim(S_2) = 0.74$, $sim(S_3) = 0.45$, $sim(S_4) = 0.55$, and $sim(S_5) = 0.68$. In this case, the services are ranked $S_1 > S_2 > S_5 > S_4 > S_3$. Service S_1 outperforms S_2 and S_5 because the cost factor carries a high weight. The weight of the cost factor was obviously larger in Z than in W (the weight vector mined by the RS), but the weights of availability and honesty were not distinguished in Z and W . When considering the cost factor, the availability weight was slightly higher than that of honesty, consistent with real production, and confirming the higher objectivity of mining the weights from historical data over relying on user evaluations. In the experiment, Service S_2 outperformed Service S_1 because the price, availability, and honesty weights were all adjusted to make it more appealing.

5.2 Comparison of Selection Success Rate

In traditional information retrieval, the performance of retrieval algorithms is commonly estimated by the *precision*. In the present experiment, the performance of the selection algorithm was analyzed by a variant of precision called the *success rate*. With the aim of recommending optimal services to users, the selection success rate is defined as follows:

$$\text{success rate} = \frac{\text{number of accepted recommendations}}{\text{total number of recommendations}} \times 100\% \quad (15)$$

This experiment measured the success rates of three selection methods: the equal-weight selection method (in which the conditional attribute weights are evenly distributed and the QoS value is neither updated nor corrected), the user fixed-weight selection method (in which the weights are preset by the user, and the QoS value is neither updated nor corrected) and the hybrid fixed-weight selection method (the proposed method, in which the user-set weights are combined with the rough-set weights). The QoS value in the hybrid method is adjusted by honesty detection and an information correction mechanism. Simulations were executed on 20 manufacturing operations, each considered by 100 users for service selection. The success rates obtained in the comparison experiment are plotted in Fig. 1.

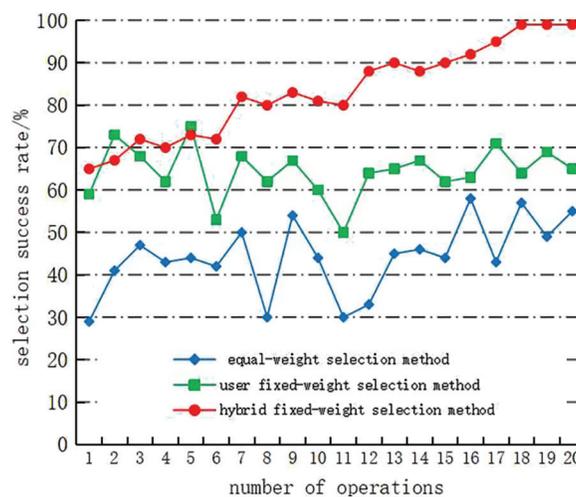


Figure 1: Precision comparison of three selection methods

As shown in Fig. 1, the user fixed-weight selection method was more successful than the equal-weight service-selection method in most cases, indicating the importance of considering the users' expectation and their preferred evaluation factors. As the equal-weight selection method does not consider the weight differences between the QoS performance indicators, its success rate was relatively low and unstable. Meanwhile, the user fixed-weight selection method is blinded to some extent by the preferences of the weight setter, so its success rate fluctuated greatly. Initially, the hybrid fixed-weight selection method did not outperform the user fixed-weight selection method, because it lacked a historical record in the earliest stages. As the historical database extended, the success rate of the hybrid method gradually increased, indicating that the weights became increasingly more objective. The stability of the success rate also improved over time. The proposed method clearly outperformed the other two methods in terms of selection success rate.

5.3 Dishonesty Evaluation Experiment

For the dishonesty evaluation, we simulated 100 randomly selected services evaluated by up to 200 dishonest users. The selection success rates of the three selection methods are compared in Fig. 2.

As shown in Fig. 2, increasing the number of dishonest users reduced the success rates of the equal-weight and user fixed-weight selection methods, because neither method adopts an honesty information detection and correction mechanism. The equal-weight selection method was especially sensitive to

dishonest evaluations, because its honesty weight was relatively large. The hybrid fixed-weight selection method introduces a QoS information correction mechanism, and collects the user evaluation data by a monitor placed on the cloud manufacturing service platform. The low quality of their evaluation data reduces the credibility of dishonest users; accordingly, these users score very lowly in the QoS honesty calculation and their evaluation information will be filtered out. Therefore, increasing the number of dishonest users barely affected the success rate of the hybrid fixed-weight selection method.

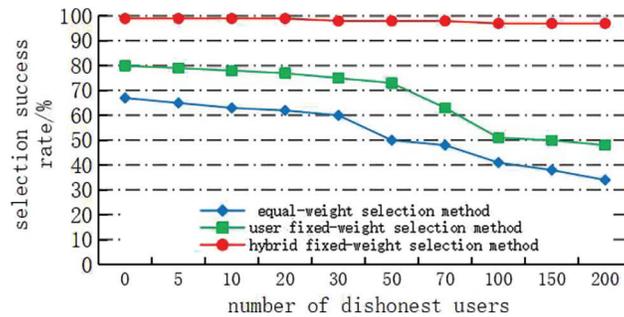


Figure 2: Comparison of selection success rates when services are evaluated by dishonest users

6 Conclusions and Future Work

This paper proposed a QoS-aware cloud service optimization model in the cloud-manufacturing environment. The study established the QoS evaluation indicators, information-aware models, and the quantification methods suitable for cloud-manufacturing businesses, and provided an honesty information correction method that filters out dishonest evaluations. The weights discovered by the variable precision RS, which mined the evaluation indicators from historical data, providing a comprehensive performance ranking of service quality. The manufacturing cloud service optimization algorithm thus provides a quantitative reference for service selection. In experimental simulations, this method recommended the optimal services and effectively reduced the impact of dishonest users on the selection results. In the next step, we will research the combination technology of manufacturing cloud services, conduct in-depth studies on the task decomposition, and improve the functionality of the developed system prototype.

Acknowledgement: I would like to express my gratitude to all those who have helped me during the writing of this thesis.

Funding Statement: This study has been supported by the National Natural Science Foundation, China (Grant No: 61602413, Jianwei Zheng, <https://www.nsf.gov.cn>), and the Natural Science Foundation of Zhejiang Province (Grant No: LY15E050007, Wenlong Ma, <http://zjnsf.kjt.zj.gov.cn/portal/index.html>).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] B. H. Li, L. Zhang, S. L. Wang, F. Tao and X. D. Chai, "Cloud manufacturing: A new service-oriented manufacturing model," *Computer Integrated Manufacturing Systems*, vol. 16, no. 1, pp. 1–7, 2010.
- [2] D. Bernbach, "Quality of cloud services: Expect the unexpected," *IEEE Internet Computing*, vol. 21, no. 1, pp. 68–72, 2017.
- [3] E. J. Ghomi, A. M. Rahmani and N. N. Qader, "Cloud manufacturing: Challenges, recent advances, open research issues, and future trends," *International Journal of Advanced Manufacturing Technology*, vol. 102, pp. 3613–3639, 2019.

- [4] M. Koehler and S. Benkner, "Design of an adaptive framework for utility-based optimization of scientific applications in the cloud," in *Proc. of the 2012 IEEE/ACM Fifth Int. Conf. on Utility and Cloud Computing*. IEEE Computer Society, Chicago, IL, USA, pp. 303–308, 2012.
- [5] N. Phaphoom, X. Wang, S. Samuel, S. Helmer and P. Abrahamsson, "A survey study on major technical barriers affecting the decision to adopt cloud services," *Journal of Systems and Software*, vol. 103, pp. 167–181, 2015.
- [6] Z. ur Rehman, O. K. Hussain, F. K. Hussain, E. Chang and T. Dillon, "User-side QoS forecasting and management of cloud services," *World Wide Web*, vol. 18, no. 6, pp. 1677–1716, 2015.
- [7] X. Zheng, L. D. Xu and S. Chai, "Qos recommendation in cloud services," *IEEE Access*, vol. 5, no. 5, pp. 5171–5177, 2017.
- [8] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu and J. Wang, "QoS ranking prediction for cloud services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1213–1222, 2013.
- [9] J. A. Alzubi, R. Manikandan, O. A. Alzubi, I. Qiqieh, R. Rahim *et al.*, "Hashed needham schroeder industrial IoT based cost optimized deep secured data transmission in cloud," *Measurement*, vol. 150, no. 1, pp. 1–8, 2020.
- [10] W. Sun, G. C. Zhang, X. R. Zhang, X. Zhang and N. N. Ge, "Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30803–30816, 2021.
- [11] D. Serrano, S. Bouchenak, Y. Kouki, T. Ledoux, J. Lejeune *et al.*, "Towards qos-oriented sla guarantees for online cloud services: Cluster, Cloud and Grid Computing (CCGrid)," in *2013 13th IEEE/ACM Int. Symp. on IEEE*, Delft, Netherlands, pp. 50–57, 2013.
- [12] H. Wu, K. Yue, C. H. Hsu, Y. Zhao, B. Zhang *et al.*, "Deviation-based neighborhood model for contextaware QoS prediction of cloud and IoT services," *Future Generation Computer Systems*, vol. 76, no. 10, pp. 550–560, 2017.
- [13] E. kodhai, K. S. V. Divakar, R. Natarajan, A. C. Allwin, P. Yellamma *et al.*, "Managing the cloud storage using deduplication and secured fuzzy keyword search for multiple data owners," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 14, pp. 563–565, 2018.
- [14] W. Sun, X. Chen, X. R. Zhang, G. Z. Dai, P. S. Chang *et al.*, "A Multi-feature learning model with enhanced local attention for vehicle re-identification," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3549–3560, 2021.
- [15] Y. Pan, S. Ding, W. Fan, J. Li and S. Yang, "Trust-enhanced cloud service selection model based on QoS analysis," *PloS One*, vol. 10, no. 11, pp. 1–19, 2015.
- [16] O. A. Alzubi, J. A. Alzubi, A. Al-Zoubi, M. A. Hassonah and U. Kose, "An efficient malware detection approach with feature weighting based on harris hawks optimization," *Cluster Computing Journal*, vol. 25, no. 4, pp. 2369–2387, 2022.
- [17] Z. Zhou, W. Xu, D. T. Pham and C. Ji, "QoS modeling and analysis for manufacturing networks: A service framework," in *Proc. of the 7th IEEE Int. Conf. on Industrial Informatics*, New York, N.Y., USA, IEEE, 2009.
- [18] Y. Liu, A. H. Ngu and L. Z. Zeng, "QoS computation and policing in dynamic Web service selection," in *Proc. of the 13th Int. Conf. on World Wide Web*, New York, N.Y., USA, ACM, pp. 17–22, 2014.
- [19] S. S. Yau and Y. Yin, "QoS-Based service ranking and selection for service-based system," in *Proc. of IEEE Int. Conf. on Services Computing*, Washington, D.C., USA, IEEE, pp. 56–63, 2011.
- [20] F. Tao, D. Zhao and L. Zhang, "Resource service optimal-selection based on intuitionistic fuzzy set and non-functionality QoS in manufacturing grid system," *Knowledge and Information System*, vol. 25, no. 1, pp. 185–208, 2010.
- [21] J. Xu, X. Du, W. Cai, C. Zhu and Y. Chen, "MeURep: A novel user reputation calculation approach in personalized cloud services," *PloS One*, vol. 14, no. 6, pp. 1–15, 2019.
- [22] L. Xiong, "Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.
- [23] S. Wang, Z. Zheng, Z. Wu, M. R. Lyu and F. Yang, "Reputation measurement and malicious feedback rating prevention in web service recommendation systems," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 755–767, 2015.

- [24] M. H. Al-Adhaileh and F. W. Alsaade, "Detecting and analysing fake opinions using artificial intelligence algorithms," *Intelligent Automation & Soft Computing*, vol. 32, no. 1, pp. 643–655, 2022.
- [25] O. A. Wahab, J. Bentahar, H. Otrok and A. Mourad, "Towards trustworthy multi-cloud services communities: A trust-based hedonic coalitional game," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 184–201, 2018.
- [26] O. A. Wahab, J. Bentahar, H. Otrok and A. Mourad, "A survey on trust and reputation models for Web services," *Decision Support Systems*, vol. 74, no. 6, pp. 121–134, 2015.
- [27] E. Yang, Z. Yong, W. Liu, Y. Liu and S. Liu, "A hybrid approach to placement of tenants for service-based multi-tenant SaaS application," in *Proc. of Asia-Pacific Services Computing Conf.*, Jeju, Korea, 2011.
- [28] E. Al-Masri and Q. HMahmoud, "Discovering the best web service," in *Proc. of the 16th Int. World Wide Web Conf.*, New York, N.Y., USA, IEEE, pp. 1257–1258, 2007.
- [29] A. A. Movassagh, J. A. Alzubi, M. Gheisari, M. Rahimi, S. K. Mohan *et al.*, "Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model," *Journal of Ambient Intelligence Humanized Computing*, vol. 3, no. 3, pp. 1–19, 2021.
- [30] Y. Chen and Y. Chen, "Feature subset selection based on variable precision neighborhood rough sets," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 1–12, 2021.