Skyline Service Selection Approach based on QoS Prediction

Yan Guo
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Haidian, Beijing, China
E-mail: guoyan@bupt.edu.cn

Shangguang Wang*
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Haidian, Beijing, China
E-mail: sgwang@bupt.edu.cn
*Corresponding author

Kok-seng Wong
School of Software
Soongsil University
Seoul, South Korea
E-mail: kswong@ssu.ac.kr

Myung Ho Kim
School of Software
Soongsil University
Seoul, South Korea
kswong@ssu.ac.kr
E-mail: kmh@ssu.ac.kr

Abstract: The Internet currently hosts a large number of Web services with highly volatile quality of service (QoS), which makes it difficult for users to quickly access highly reliable online services. Hence, the selection of the optimal service composition based on fast and reliable QoS has emerged as a challenging and popular problem in the field of service computing. In this paper, we propose a service selection approach based on QoS prediction. We consider historical QoS information as time series and predict QoS values using the autoregressive integrated moving average model, which can provide more accurate QoS attribute values. We then calculate the uncertainty in the prediction results using an improved coefficient of variation to prune redundant services. In order to downsize the search space, we employ Skyline computing to prune redundant services and perform Skyline service selection by using 0-1 mixed-integer programming. Experimental results based on real-world dataset showed that our approach yields satisfactory performance in terms of reliability and efficiency.

Keywords: service selection; QoS prediction; autoregressive integrated moving average model; Skyline service.

Reference to this paper should be made as follows: Y. Guo, S. Wang, K. Wong, and M. Kim (2016), 'Skyline Service Selection Approach based on QoS Prediction,' *International Journal of Web and Grid Services*, Vol. , No. , pp.xx–xx.

Biographical notes:. Yan Guo is a first-year Master's student at Beijing University of Posts and Telecommunications, and will be a PhD student next year. Her research interests include service computing and edge computing.

Shangguang Wang is an associate professor at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT). He received his Ph.D. degree at BUPT in 2011. Dr. Wang is Vice Chair of IEEE Computer Society Technical Committee on Services Computing, General Chair of ICCSA 2016, Program Chair of the 2014 IOV, and Program Chair of the 2014 SC2. He has published more than 100 papers

Kok-Seng Wong obtained his PhD in Computer Science from Soongsil University, South Korea in 2012. He is currently working as an Associate Professor in the School of Software at Soongsil University. His research interests include security and data privacy, secure computation, cryptographic protocols and cloud computing.

Myung Ho Kim received his PhD in computer science from POSTECH in 1995. He has been a professor at the School of Software at Soongsil University since September 1995. He is now chair of the School of Software. He specializes in business intelligence, Internet security, and distributed computing.

1 Introduction

A Web service is an application that is platform independent, programmable, self-contained, features low coupling, and has certain other characteristics (Alonso et al., 2010). With the development of the Internet, the demands of users are becoming ever more complex, particularly as they pertain to Web services. Therefore, it is useful to combine the available Web services into a more powerful composition service to meet users' needs (Wang et al., 2016). On the contrary, with the rapid development of Web service technology, the number of Web services deployed on networks is increasing rapidly (Wang et al., 2015). A large number of services with the same or similar functional attributes but different non-functional attributes have been proposed, which makes it even more difficult for users to quickly obtain composite services with high reliability. Therefore, the selection and combination of Web services on different network platforms has become a key issue in Web service technology.

According to the service-oriented architecture (SOA) paradigm, a composition service is composed of a series of abstract services (service classes). In the composition process, a specific Web service (the candidate service) is selected from each service class (Canfora et al., 2008). The selection of a Web service not only requires considering whether the functional attributes of the service meet the users' needs, but also ensuring whether the quality of service (QoS) and other nonfunctional attributes of Web services satisfy users (Commonly used QoS attributes include response time, cost, throughput, credibility, reliability, and availability) (Wang, 2011).

Although current SOAs can support Web service registration, discovery, and composition, there remain many challenges in effectively integrating large numbers of Web services into a reliable, new service according to users' QoS requests (Wang et al, 2013).

This problem has attracted attention from industry and academia, especially with regard to the optimization of service selection approaches based on QoS in service composition (Wang et al., 2012).

Commonly used Web service selection approaches can be divided into approaches based on the function and those based on QoS (Dai, 2013). The former chooses appropriate services to meet the functional request, which is the basic requirement of service composition; the latter chooses appropriate services to satisfy request relating to QoS. In recent years, research has intensified on Web service selection approaches based on QoS. However, the number of the Web services with the same or similar functions ever growing, and the number of the candidate services from each service class is increasing as well. Hence, there are a large number of possible combinations of services. This is a typical NP-hard problem (Hwang et al., 2008) (Liu et al., 2010).

Although existing service selection approaches have achieved satisfactory results and played an important role in promoting applications of composite Web services, three problems persist:

1) Limited historical QoS records. In previous studies, the values of QoS attributes are generally expressed as the arithmetic mean of historical records. If there is a tendency in the sequence of historical QoS records, it is inaccurate to denote further QoS values based only on the mean of historical QoS records, and this influences the success rate of the final Web service composition.
2) Ignoring QoS uncertainty. Due to dynamic nature of Web service environments, the true QoS values of some candidate services deviate from the aggregate QoS attribute values (Wang, 2011), which lowers the reliability of service selection, and even leads to failure of service selection.
3) Poor real-time performance. Existing selection approaches usually focus excessively on service selection optimization algorithms to reduce computation time while ignoring the fundamental factor in high time cost (i.e., the exponential increase in the number of candidate services). When service users face a large number of candidate services with the same functionality but different QoS, many excellent service selection optimization algorithms still consume a large amount of computation time.

In view of these three problems, we propose a service selection approach based on QoS prediction. The main contributions of this paper can be summarized as follows:

1) By QoS prediction based on historical QoS records, we mine useful information that can help us select services based on these records.
2) Based on QoS prediction, we propose a fast and reliable Skyline service selection approach. This approach yields impressive performance from three aspects: providing more accurate forecasting of QoS information for Web service selection to improve the success rate of Web service composition, reducing uncertainty in service selection to improve the reliability of selection composition, downsizing the scale of the selection space to improve the real-time of service selection.
3) Based on a real-world dataset, we conducted a series of experiments to compare our approach with two others in terms of reliability and efficiency. The experimental results showed that our approach can find the best Skyline service selection solution in lesser computation time.

The rest of this paper is organized as follows: Section 2 introduces the background of service composition. Section 3 describes the proposed service selection approach. Section 4 details our experiments, and Section 5 contains our conclusions.

2 Background

2.1. Related concepts

Service composition technology forms the core technology of service-oriented architecture and service-oriented computing, and can quickly meet the requirements of complex, dynamic, and inter-organizational businesses. It consists of the following basic concepts:

- A service candidate is a service that can satisfy a user's particular functional demand.
- A service class is a set composed of multiple service candidates with the same functions but different non-functional attribute values.
- Service selection involves choosing the most suitable service candidate(s) to meet the functional and QoS requirements of users from one or more service classes.

In order to better understand the above three concepts, we give the corresponding character representation. A composition service process $\mathbb{S}$ contains n service classes, i.e., $\mathbb{S}=\{S_1,S_2,\cdots,S_n\}$. $S_i(0<i\le n)$ represents a concrete service class that contains $l(l>1)$ candidate services with different QoS values, i.e., $S_i=\{s_{i1},s_{i2},\cdots,s_{il}\}$. $s_{ij}(0<j\le l)$ represents a concrete service candidate. Service selection then involves selecting a s_{ij} from each S_i to integrate a composition service under the global constraints of QoS.

The QoS is the commonest index to measure the merits of multiple candidate services with the same functionality in service selection based on non-functional attributes (Ma et al., 2016) (Wang et al., 2015). The QoS attributes of Web services are generally divided into two categories: active QoS attributes and negative QoS attributes (Lin et al., 2011).

The character representation is as follows: candidate service s has r QoS attributes, and the QoS vector of s is $Qs=\{q_1(s),q_2(s),\cdots,q_r(s)\}$, where $q_k(s)(0<k\le r)$ represents the k-th attribute value of s. The QoS attribute value of the composite service is obtained by aggregating the corresponding attribute values of candidate services selected from the respective service classes. Similarly, the QoS vector of composition service S is $QS=\{q_1(S),q_2(S),\cdots,q_r(S)\}$. The QoS aggregation functions in the sequential composition model have been shown in (Wang et al., 2014). Other models (e.g., parallel, conditional, and loops) can be transformed into the sequential model (Jang et al., 2006).

In service selection, it is difficult to compare and sort services, for a service has several QoS attributes. A utility function is designed to map the vector of QoS values into a real value to compare and sort candidate services and composition services. The utility functions (Wang et al., 2016) of a candidate service and a composition service can be computed as follows in the sequential composition model:

$$U(s)=\sum_{k=1}^{\alpha}\frac{Q_{i,k}^{\max}-q_k(s)}{Q_{i,k}^{\max}-Q_{i,k}^{\min}}.\omega_k+\sum_{k=1}^{\beta}\frac{q_k(s)-Q_{i,k}^{\min}}{Q_{i,k}^{\max}-Q_{i,k}^{\min}}.\omega_k \tag{1}$$

$$U(S)=\sum_{k=1}^{\alpha}\frac{Q_k^{\max}-q_k(S)}{Q_k^{\max}-Q_\alpha^{\min}}.\omega_k+\sum_{k=1}^{\beta}\frac{q_k(S)-Q_k^{\min}}{Q_k^{\max}-Q_k^{\min}}.\omega_k \tag{2}$$

$$\begin{cases} Q_k^{\max} = \sum_{i=1}^{n} Q_{i,k}^{\max} & (Q_{i,k}^{\max} = \max_{\forall s_{ij} \in s_i} q_k(s_{ij})) \\ Q_k^{\min} = \sum_{i=1}^{n} Q_{i,k}^{\min} & (Q_{i,k}^{\min} = \min_{\forall s_{ij} \in s_i} q_k(s_{ij})) \end{cases} \tag{3}$$

where α is the number of negative QoS attributes and β is the number of positive QoS attributes ($\alpha + \beta = r$), $\omega_k (\sum_{k=1}^{r} \omega_k = 1)$ is the weight of the *k*-th QoS attribute representing the preference of users, $Q_{i,k}^{\max}$ is the maximum value of the *k*-th QoS attribute in all candidate services of the *i*-th service class, $Q_k^{\max}$ is the maximum value of the *k*-th QoS attribute in all composition services $\mathbb{S}$, and $Q_{i,k}^{\min}$ and $Q_k^{\min}$ are similar.

2.2. Related work

A number of service selection approaches have been proposed in the literature. Here, we only review some notable ones.

In early research in the field, many researchers focused on QoS-based service selection. A few focused on improving algorithms to reduce the complexity of service composition. Traditional optimization algorithms, such as the exhaustive algorithm (Gao et al., 2006) and the greedy algorithm (Ding et al. 2009), and traditional heuristic algorithms, such as the genetic algorithm (Tang and Ai, 2010) and particle swarm optimization (Li and Huang, 2010) (Wang et al, 2013), have been researched very thoroughly. When the number of candidate services is small, these approaches yield good performance in terms of real-time results and reliability. However, the computational complexity and cost of these approaches increase exponentially with growth in the number of services.

To improve the efficiency of service selection, many near-to-optimal service selection approaches have been proposed. (Wan et al., 2008) proposed an efficient divide-and-conquer algorithm that handled complex control flows in an integrated way without separating and merging multiple execution paths, and divided the original service into several smaller services that were then solved separately by a recursive branch-and-bound algorithm. (Alrifai et al., 2009) combined global optimization with local selection techniques to both handle global QoS requirements and real-time performance. It decomposed global QoS constraints into local constraints, and then found the best Web services satisfying the local constraints by using distributed local selection.

Compared with the traditional approaches, these near-to-optimal approaches reduce selection cost, and are appropriate for applications with dynamic and real-time requirements, but focus excessively on the optimization of the selection algorithms themselves to the neglect of the basic impact factor: rapid growth in the number of services.

In contrast to the above work, some research in the area has focused on techniques to shorten the time needed for service selection or improve the reliability of the results of this selection. One such effective technique is the Skyline service. (Alrifai et al., 2010) proposed an approach based on the notion of a Skyline to select services effectively and efficiently. The approach reduced the number of candidate services by considering the dominance relationships among Web services based on their QoS attributes, and showed how to consider only a subset of Skyline services for composition. (Hiratsuka et al., 2011) proposed two approaches to reduce the cost of service selection based on QoS. One involved the gradual updating of Skyline services and the other involved the removal of service combinations by grouping. The former reduced the number of services by using Skyline services, whereas the latter reduced the number of combinations by eliminating

services with similar QoS values into two representative services. The two approaches yield low computational cost in service selection composition.

Except for Skyline services, the QoS dependence of services is another often considered technique. (Barakat et al., 2012) focused on QoS correlation and presented a Correlation-aware Service Selection Model that can handle the QoS dependencies of services and improves the quality of composition. This model introduced correlation-aware pruning techniques that can eliminate uninteresting compositions from the search space before selection to reduce selection cost. This study also accounted for service quality dependencies in selection while performing pruning prior to and during selection to improve performance. (Feng et al., 2013) considered QoS-aware service composition in the presence of service-dependent QoS and proposed a formal model that captures both partial and full dependencies as well as a novel approach that can dynamically refine the composed workflow in light of QoS dependencies as well as user-provided topological and QoS constraints.

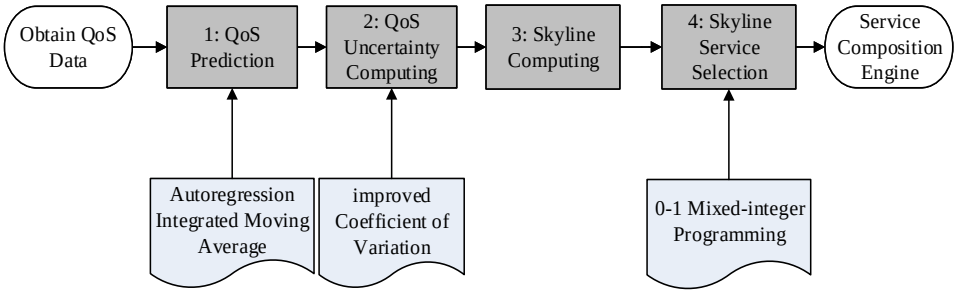
In addition to these two commonly used techniques, (Wang, 2011) employed the cloud model to compute uncertainty of candidate services by transforming quantitative QoS values into qualitative concepts, pruned redundant services with high uncertainty, and used mixed-integer programming (MIP) to select the optimal services. (Sun et al., 2014) not only considered the QoS uncertainty of Web services, but also paid greater attention to downsizing the solution spaces of the service selection process. They adopt the concept of entropy from information theory and variance theory to filter redundant services with low reliability, thus reduce the number of unreliable candidate services, and improve the reliability and the efficiency of service selection. Both approaches consider the uncertainty of services and prune redundant services by computing uncertainty. These approaches reduce the computational time and improve the reliability of selection results.

However, all these studies calculate the values of QoS attributes using the arithmetic mean of the historical records. This renders the QoS value inaccurate and influences the success rate of service composition. Therefore, we propose a service selection approach based on QoS prediction where we calculate the values of QoS attributes using predicted values containing tendency information. This improves the reliability of our approach in comparison with the above techniques.

3 Our Service Selection Approach

As shown in Figure 1, the approach proposed in this paper contains four phases. The first phase is QoS prediction, where we create a model by analyzing the time series data of historical QoS records based on the autoregressive integrated moving average (ARIMA) model of time series analysis, and then use the model to predict the QoS attributes values of candidate services. The second phase is QoS uncertainty computing, where we adopt variance theory to compute the uncertainty of services and filter candidate services with high uncertainty. The third phase is Skyline computing, where we adopt the Skyline service to downsize the solution space further and improve real-time performance. The final phase is Skyline service selection, where we use the 0-1 MIP algorithm to find the service composition with the higher reliability and shorter computation time.

Figure 1 Procedures of our approach



3.1 QoS Prediction

To improve the reliability of service composition, we adopt the ARIMA model to predict QoS values in the near future. Based on a large number of historical QoS records, we use the ARIMA model to fit data, build a QoS prediction model, and then predict QoS values.

3.1.1. ARIMA model

Time series forecasting approaches are widely used in many areas, such as market analysis, econometrics, financial mathematics, information processing, and so on. The basic idea is to regard the data sequence formed over time as a random sequence, and create a certain mathematical model to describe the sequence that can predict future values from past and present values of the time series.

ARIMA is a commonly used and effective approach in time series prediction (Janacek, 1990). It transforms non-stationary time series into stationary time series, and establishes a regression model for dependent variables based only on its lag value and the present value, and the lag value of a random error term. Its specific form can be expressed as ARIMA (*p, d, q*), where *p* indicates the order of the autoregressive process, *d* indicates the difference of the order, and *q* represents the order of the moving average process.

The ARIMA model is defined for stationary time series, hence, we need to preprocess the original data. Figuratively, a time series $\{x_t\}$ must be turned into a stationary series $\{w_t\}$ first, following which the ARMA (*p, q)* model of $\{w_t\}$ is established. We can obtain the ARIMA (*p, d, q*) model of $\{x_t\}$ through a transformation. In general, the stationary time series refers to a wide stationary process (Janacek, 1990). In a loose sense, the stationary time series refers to time series whose average variance and autoregression function almost do not change with time.

Definition 1 **(The ARMA model):** The mathematical model with the following structure is called the ARMA model:

$$\begin{cases} x_t = \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_p x_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \\ \varphi_p \neq 0, \theta_q \neq 0 \\ E(\varepsilon_t) = 0, Var(\varepsilon_t) = \sigma_\varepsilon^{\,2}, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \\ E(\varepsilon_t x_s) = 0, \forall s < t \end{cases} \tag{4}$$

In service selection scenarios, $\{x_t\}$ $(t = 1, 2, \cdots\cdots)$ is a historical QoS record in t time periods, $\{\varepsilon_t\}$ is a white noise sequence that cannot be observed and contains the tendency of the historical QoS records time series, p and q are estimated according to the autocorrelation function (ACF) and the partial autocorrelation function (PACF) of the

historical QoS records time series, respectively, and $\{\varphi_p\}$ and $\{\theta_q\}$ are determined by matching historical QoS records.

3.1.2. ARIMA model for QoS prediction

The basic procedures involved in building the ARIMA model are given below:

1) Data preprocessing

According to its scatter plot, ACF, PACF, and unit root test, we test the variance, tendency, and the seasonal variation law of the time series, and identify the stationarity of the sequence. If the time series is not stationary, we need to use differencing approach until it is smooth, and then use ARMA to model the stationary sequence. The times of difference is the value of *d* in the model ARIMA (*p, d, q*).

2) Model identification

We determine model types according to the ACF and the PACF. This process is used to estimate the orders of autoregression and the moving average; therefore, the model identification process is also called the order estimation process.

Table 1 Model identification

Model	ACF	PACF
AR(p)	tailing	p-order truncated
MA(q)	q-order truncated	tailing
ARMA(p,q)	tailing	tailing

As shown in Table 2, if the PACF of a stationary sequence is truncated and the ACF is tailing, it can be concluded that the sequence fits the AR model; if the PACF of a stationary sequence is tailing and the ACF is truncated, the sequence fits the MA model; if both the PACF and ACF are trailing, the sequence fits the ARMA model.

3) Estimate the parameters

In the previous step, we obtain the values of p and q in ARMA(*p, q*). The parameters that need to be fitted are $\{\varphi_p\}$ and $\{\theta_q\}$. According to the historical QoS sequence of the service, we can fit parameters $\{\varphi_p\}$ and $\{\theta_q\}$ and obtain the ARIMA model that can predict short-term QoS values.

4) Prediction and analysis

Using the model fitted for QoS prediction, we predict QoS attributes values in the *(t*+1*)*-th time period, and obtain the predicted value of each historical record value. Hence, in this paper, we can estimate the reliability of the prediction result using the deviation between the predicted historical QoS values and real historical QoS records.

To explain this QoS prediction model more clearly, we provide a prediction example of the QoS value of a Web service. The information pertaining to services was taken from the WS-Dream dataset[1]. For ease of explanation, we only consider one QoS attribute of a Web service.

We first drew the time series chart (Figure 2), the ACF chart (Figure 3), and the PACF chart (Figure 4) of historical QoS records, and performed the unit root test on the data. The

[1] http://www.wsdream.net

result of the test was $H=1$ and significance p was 0.001. From this result, we concluded that the sequence was stationary, and we did not need to difference the data. In this case, the value of parameter d was 0. Then, according to the ACF and the PACF, we could identify that $p=1$ and $q=1$; hence, the model was ARIMA (1, 0, 1). We then used the first 249 data items to predict the 250th item, and performed the white noise test to the residual sequence; the result was $H=0$, which meant this residual sequence was random: that is to say, this model and the prediction results were reliable.

Figure 2 Time series chart of historical QoS records

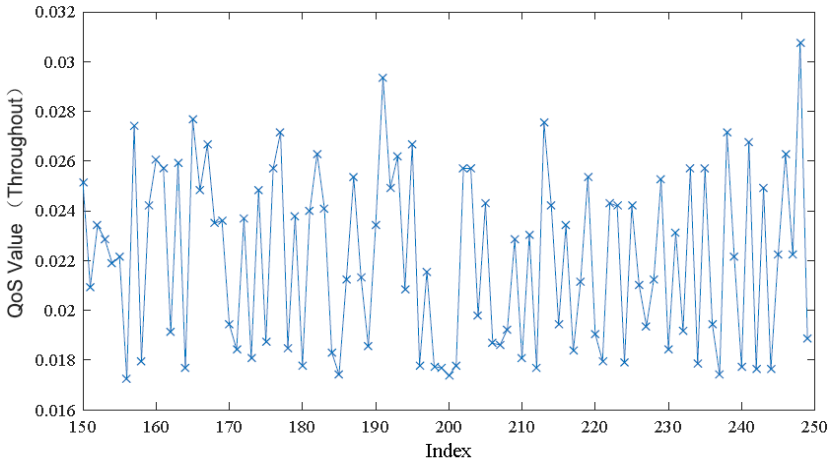


Figure 3 Autocorrelation function graph

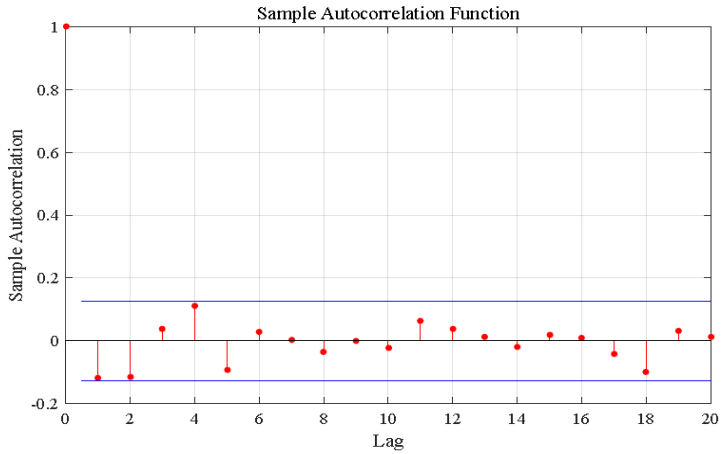


Figure 4 The partial autocorrelation function

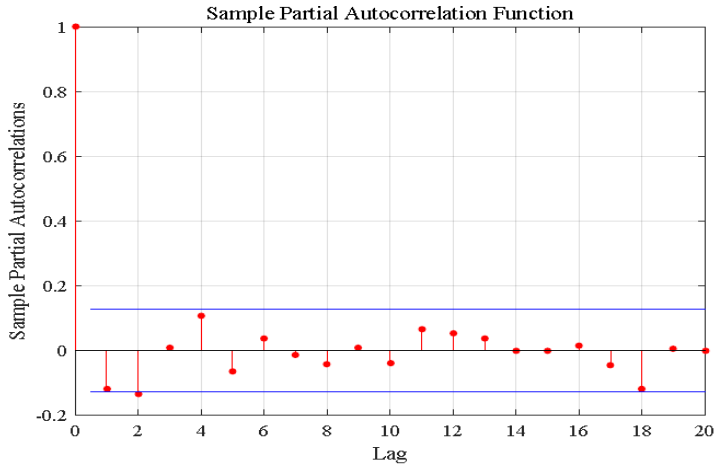
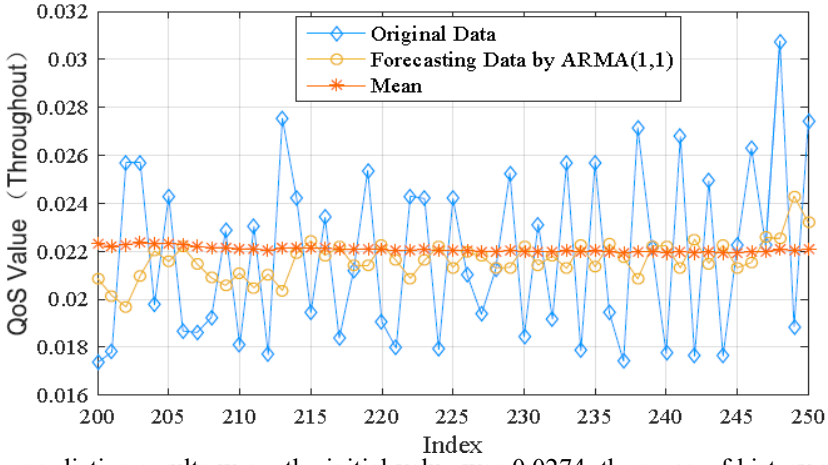


Figure 5 Comparison of prediction result



The prediction results were: the initial value was 0.0274, the mean of history records was 0.0222, and the predicted value was 0.0232. We can see that the ARIMA model's predicted value was better than the mean value. Figure 5 is part of the comparison image of the original example data, the predicted QoS values and the mean values, and shows that the value predicted by the ARIMA model was better than the mean value in most cases.

3.2 Uncertainty Computing

Although we tried to render research on the information in historical QoS records in the QoS prediction phase in as detailed a manner as possible, the future situation of candidate services is still likely to deviate from the expected state, and the actual effect of service composition is likely to deviate from the prediction effect. Therefore, service selection is likely to face the risk of failure. This is due to the dynamic Web environment (Wang et al., 2014) and forecast uncertainty.

To reduce the uncertainty of service selection, we hope to compute forecast uncertainty using the deviation between the predicted historical QoS values and real historical QoS records. We thus improve the coefficient of variation, a statistic used to measure variation in data in information theory, to reflect the uncertainty of our forecast model in this paper.

***Definition 2* (The improved Coefficient of Variation):** Let X be a random variable, and $\{x_1, x_2, \cdots, x_t\}$ be the range of X. The improved coefficient of variation (*ICV*) can be expressed by the following:

$$ICV = \tilde{S} / \overline{X} \times 100\% \tag{5}$$

$$\begin{cases} \overline{X} = \dfrac{1}{t}\sum_{i=1}^{t} x_i \\ \tilde{S} = \sqrt{\dfrac{1}{t-1}\sum_{i=1}^{t}(x_i - \overline{x_i})^2} \end{cases} \tag{6}$$

In service selection scenarios, x_i represents the historical QoS attribute record value in the *i*-th time period, t represents the number of time periods, $\overline{X}$ is the mean of the history records, $\overline{x_i}$ represents the QoS predicted value in the *i*-th time period, and $\tilde{S}$ is the improved standard deviation. Compared with the natural coefficient of variation (Sun et al., 2014), $x_i - \overline{X}$ is replaced by $x_i - \overline{x_i}$. The sum of squared residuals is used to reflect the uncertainty of the predicted value in *ICV*. By using *ICV*, we can find and filter the candidate services with high uncertainty.

3.3 Skyline Computing

Following the computation of QoS uncertainty, we filter the service candidates with high uncertainty. However, the search space is still large, and most of the remaining candidate services are redundant. Therefore, we adopt Skyline computing (Borzsony et al., 2001) to reduce the search space further.

Skyline computing, also called Pareto optimality, extracts elements that cannot be dominated by other elements from the database (Wang et al., 2012).

***Definition 3 (Service Domination)*:** There are two candidate services s_1 and s_2, and each service has r negative QoS attributes. The *k*-th attribute value is expressed by $q_k(s_1)$ $q_k(s_2)$. We say that s_1 dominates (Pareto optimizes) s_2 if

$$q_k(s_1) \le q_k(s_2), \forall k \in \{1, 2, \cdots, r\} \tag{7}$$

$$q_k(s_1) < q_k(s_2), \exists k \in \{1, 2, \cdots, r\} \tag{8}$$

It is denoted by $s_1 \succ s_2$.

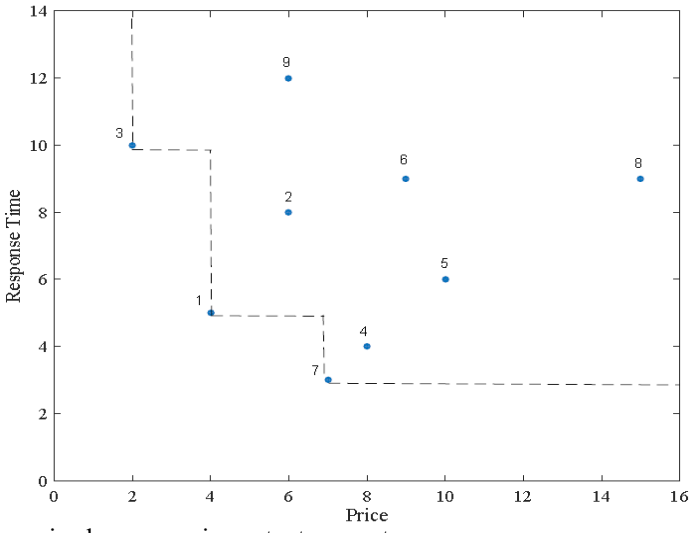
In many cases, the domination relationship between two services is cannot be judged. The Skyline service is a set of such services; it contains all services that cannot be dominated by other services, and there is no better or worse in terms of services in this set.

***Definition 4 (Skyline Service)*:** There is a service class $S = \{s_1, s_2, \cdots, s_l\}$; the Skyline service of S is the set of services that cannot be dominated by any other candidate service. That is to say,

$$SkylineS = \{s_i \in S \mid \nexists s_j \in S : s_j \succ s_i\} \tag{9}$$

The following figure is an example of Skyline services. The nine points represent nine feasible services, and smaller values are preferred to larger ones. Point 2 does not belong to Skyline services because it is dominated by Point 1. Similarly, only Points 1, 3, and 7 are not strictly dominated by any other, and hence belong to Skyline services. In this example, we can see that six redundant services have been filtered through Skyline computing, and the number of candidate services has changed from nine to three.

Figure 6 An example of Skyline service



The Skyline service has a very important property:

Property 1: $\mathbb{S}=\{s_1, s_2, \cdots, s_n\}$ is the best service composition satisfying global QoS constraints, where s_i is the selected concrete service from service class S_i. Each concrete service s_i in $\mathbb{S}$ belongs to the Skyline service, denoted by $s_i \in SkylineS_i$.

Proof: Assuming that $s_i \notin Skyline$, there must be a service $\overline{s_i}$ satisfying $\overline{s_i} \in SkylineS_i$ and $\overline{s_i} \succ s_i$. Replace s_i with $\overline{s_i}$. A new service composition $\overline{\mathbb{S}}=\{s_1, s_2, \cdots, \overline{s_i}, \cdots, s_n\}$ can be obtained. Since the style of the unify function (1,2) is the order model in our approach, the QoS unify function is monotonous: that is to say, the better the QoS attribute value, the greater the utility function value. Therefore, the unify function value of $\overline{\mathbb{S}}$ must be greater than the function value of $\mathbb{S}$. However, this is contradicted with the known condition that $\mathbb{S}$ is the best service composition. Thus, the assumption does not hold, and $s_i \in SkylineS_i$.

3.4 Skyline Service Selection

QoS uncertainty computing and Skyline computing reduce a large number of redundant services, speed up the service selection process, and improve the reliability of service selection. We need to select the best Skyline service from each class under global QoS constraints.

As it is well-known that the service selection problem is a multi-objective optimization model, it can be denoted by the following mathematical model:

$$\text{Min } \{q_1(\mathbb{S}), q_2(\mathbb{S}), \cdots, q_r(\mathbb{S})\} \qquad (10)$$

$$\text{subject to } q_k(\mathbb{S}) \le (\ge) C_k,\ k=1,2,\cdots,m$$

where $r \ge 2$ is the number of optimized QoS attributes, $C_k (k=1,2,\cdots,m)$ is the set of global constraints of the *k*-th QoS attribute, and $m \le r$. In order to make the model look more concise, we only consider the minimization. Some QoS attributes need to be maximized (e.g., throughput rate), and it is easy to effect a minimization, such as taking the opposite of the maximization goals.

In general, multi-objective optimization functions rarely have an optimal solution meeting all constraints. Therefore, in this paper, we adopt the Simple Additive Weighting (SAW) model to transform this multi-objective optimization problem into a single-objective optimization problem (Alrifai et al., 2010). The SAW model contains two steps: normalization, and determining the weight.

1) Normalization

Normalization limits the values in the same range and increases the credibility of the utility function. There are many ways to normalize, such as deviation standardization and logarithmic standardization. In this paper, we use the linear conversion (Min-Max scaling) to limit QoS values to the interval [0,1].

2) Determining the weight

In general, we set the weight according to the priority of different objective functions, or the preferences of users. However, the preferences of users are very difficult to know ahead of time. Even in case they are already known, setting the weight accurately remains a difficult problem. For convenience, we think the preference of users for each target is the same: that is to say, the weight of each QoS attribute is the same.

Summing all weighted objective functions, we can obtain a comprehensive utility function to represent the overall optimization goal. Thus, multi-objective optimization is turned into single-objective optimization.

Definition 4 (QoS Utility Function): Suppose there are α negative QoS attributions and β positive QoS attributions. The overall utility value is as follows:

$$F(S)=\sum_{k=1}^{\alpha}\frac{Q_k^{\max}-\sum_{i=1}^{n}\sum_{j=1}^{l}x_{ij}\cdot\hat{q}_k(s_{ij})}{Q_k^{\max}-Q_k^{\min}}.\omega_k+\sum_{k=1}^{\beta}\frac{\sum_{i=1}^{n}\sum_{j=1}^{l}x_{ij}\cdot\hat{q}_k(s_{ij})-Q_k^{\min}}{Q_k^{\max}-Q_k^{\min}}.\omega_k \tag{11}$$

where x_{ij} is a binary decision variable representing whether a service candidate is selected; if the value of x_{ij} is 1, it indicates the corresponding candidate service s_{ij} has been selected, and has been discarded otherwise. $\hat{q}_k(s_{ij})$ ($0<k<\alpha$ or $0<k<\beta$) represents the predicted valued of the *k*-th attribute of service s_{ij}, and Q_k^{max}, Q_k^{min} can be calculated by (3).

The MIP is adopted to solve this single-objective optimization problem in this paper, and can be formulated as follows:

$$\text{Max} \quad F(S) \tag{12}$$

$$\text{subject to}\begin{cases}\sum_{i=1}^{n}\sum_{j=1}^{l}q_k(s_{ij})\cdot x_{ij}\le(\ge)C_k, 1\le k\le m\le r\\ \sum_{j=1}^{l}x_{ij}=1, 1\le i\le n, x_{ij}\in\{0,1\}\end{cases} \tag{13}$$

There are many ways to solve this model. In this paper, we use the exhaustive approach.

4 Experiment

In this section, we compare our approach, called ASMIP, with the MIP approach (Ardagna et al., 2007) and the SkylineMIP approach (Alrifai et al., 2010) based on a real-world Web service QoS dataset (Zheng et al., 2010), in terms of reliability and computation time. The experimental results showed that our approach yields impressive performance in terms of

computation time and reliability. Moreover, to analyze the performance of our approach further, we also analyzed the parameters used in our approach.

4.1 Experiment Setup

The real-world data set used in our experiments is the WS-Dream, containing approximately 2 million Web service invocation records collected from 150 service users on 10,258 Web services. Each record contains several QoS attributes. For convenience, we only considered response time and throughout in our simulation experiments.

In our experiments, the number of service classes n was 5, the number of candidate services in each service class l varied from 100 to 1,000, the number of QoS attributes r was two, the weights of the two QoS attributes were 0.5, the number of historical records of each candidate service was 250, and the number of global constraints m was 2. According to the value of ICV, we filtered 50% of the candidate services with low reliability in each class.

All experiments were conducted on the same computer with an Intel(R) Core(TM) 2.5 GHz processor, 4.0 GB of RAM, on Windows 10 and MATLAB R2015a.

4.2 Reliability

In this paper, the ARIMA forecasting model was introduced to improve the accuracy of QoS prediction and the reliability of service composition. In order to compare the reliability of the three approaches, we used the first 245 historical QoS attribute records to select a service and the 246-th historical record to compute the QoS utility values of the selected service. Reliability was defined as follows:

Definition 5 (Reliability) $\tilde{U}_{ASMIP}$ is the QoS utility value of the service composition obtained through the ASMIP approach, and $\tilde{U}_{MIP}$ is the QoS utility value of the service composition obtained through the MIP approach. The reliability of ASMIP can then be expressed by the following formula:

$$Re = \begin{cases} \tilde{U}_{ASMIP} / \tilde{U}_{MIP}, & \text{if } \tilde{U}_{ASMIP} < \tilde{U}_{MIP} \\ 1, & \text{if } \tilde{U}_{ASMIP} \geq \tilde{U}_{MIP} \end{cases} \tag{14}$$

Figure 7 Comparison results in terms of reliability

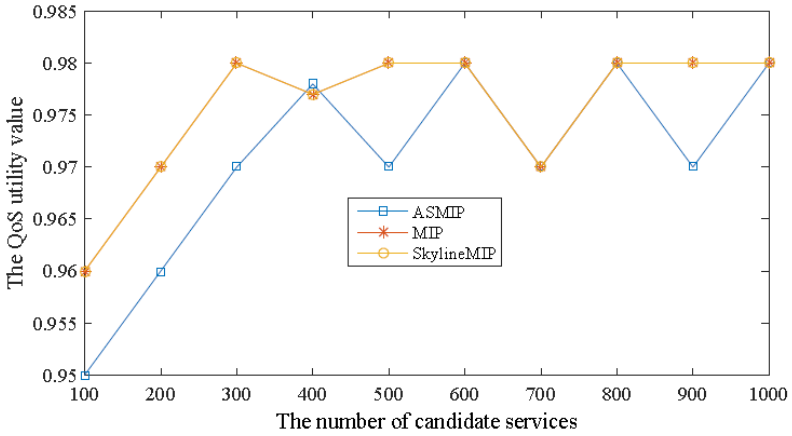


Figure 7 shows the comparison results in terms of reliability. Similar to other

approaches, our approach is highly reliable. The reliability values of our approach are near 1, which means that it is able to find the best Skyline service. That is because QoS prediction and uncertainty computing were used to improve the reliability of service selection. Although the reliability values of our approach are not all equal to 1, considering the complex dynamic Web environment, we still think this result is acceptable.

Note that in Figure 7, the reliability of the MIP and SkylineMIP approaches are similar, mainly because Skyline computing does not influence the reliability of service selection (see Property 1).

4.3 Computation Time

Existing service selection approaches usually ignore the filtering of redundant services. This not only affects the reliability of service selection, but also increases time consumption. In this subsection, we compare the computation time of these approaches.

Figure 8 Comparison results in terms of computation time

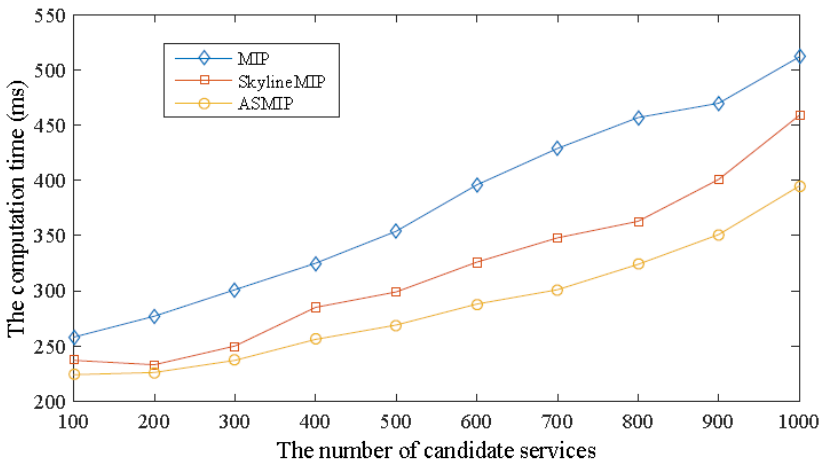


Figure 8 shows the comparison results in terms of computation time when the number of candidate services varies from 100 to 1,000. As shown in Figure 8, our approach is superior to other approaches, and its time taken is 10% and 24% shorter than MIP and SkylineMIP on average, respectively. The is because our approach filters a large number of redundant services by Skyline computing, and downsizes the search space.

In a word, as shown in Figures 7 and 8, our approach can find the best Skyline service selection solution in shorter computation time than other approaches.

4.4 Parameter Analysis of ICV

In this paper, the improved coefficient of variation (*ICV*) was used to measure the uncertainty of services. In previous experiments, 50% of the candidate services were filtered according to their *ICV* values. The ratio of filtration also affects the results of the experiment; in this subsection, we analyze the effect of the filtration ratio on performance and computation time.

Figure 9 The change in the maximum utility value with filtration ratio

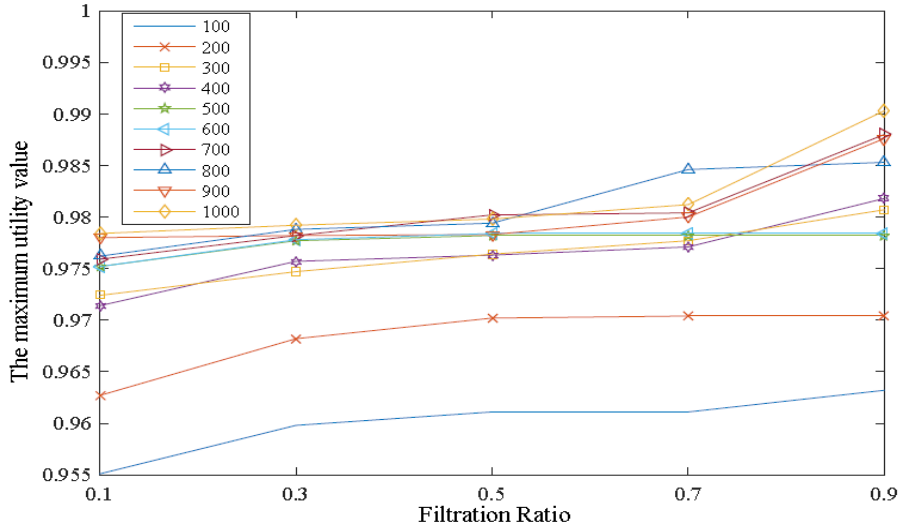


Figure 10 The change in computation time with filtration ratio

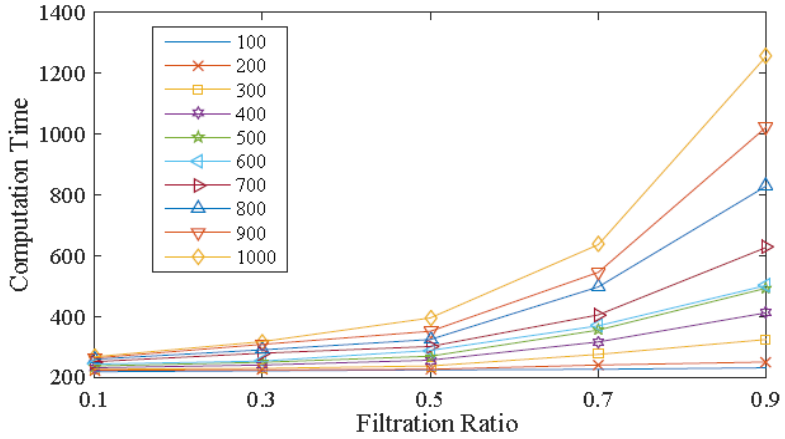


Figure 9 shows the change in the maximum utility value with the filtration ratio when the number of candidate services varies from 100 to 1,000. Figure 10 shows the change in computation time with filtration ratio when the number of candidate services varies from 100 to 1,000. As shown in these figures, with decreasing filtration ratio, the maximum utility value of the composite service gradually increases, the increments taper off, and the maximum utility value tends to be the same, except for when the number of services is 700, 900, and 1,000. The computation time increases with decreasing filtration ratio, and the greater the number of candidate services, the greater the increment. Therefore, when the ratio is 0.5, the running time is short and the maximum utility value is within the acceptable range.

5 Conclusion

Aiming at fast and efficient service selection, we proposed in this paper a Skyline service selection approach based on QoS prediction (ASMIP).

In ASMIP, the QoS attributes of services are regarded as random variables, and the historical information concerning QoS is regarded as a time series. We use the ARIMA model to predict the QoS value in the short-term future, calculate the uncertainty of the forecasting model, and then filter out Web services with high uncertainty. To further downsize the search space, Skyline computing is adopted to reduce the number of redundant services. Finally, the 0-1 MIP is used to select the optimal service composition that satisfies global QoS constraints. In order to evaluate the performance of our approach, we used a real-world dataset containing 5,825 Web services for simulation experiments to test for computation time, reliability, and parameter analysis. The results showed that the ASMIP approach can find the best Skyline service selection solution in shorter computation time than other approaches.

However, ASMIP is based on a large number of historical QoS attribute records of candidate services, and is not suitable for the selection of new services with fewer historical records. Therefore, the next study in this vein should focus on new services for more effective service selection approaches. Moreover, we need to conduct further research on the QoS prediction model to improve the reliability of service selection.

ACKNOWLEDGMENTS

This work was supported by the NSFC (61202435 and 61472047).

6 References

Alonso, G., Casati, F., Kuno, H. and Machiraju, V. (2010) 'Web Services: Concepts, Architectures and Applications': *Springer Publishing Company, Incorporated*.

Alrifai, M. and Risse, T. (2009) 'Combining global optimization with local selection for efficient QoS-aware service composition'. *Proceedings of the 18th international conference on World Wide Web (WWW 2009)*, Madrid, Spain, pp. 881-890.

Alrifai, M., Skoutas, D. and Risse, T. (2010) 'Selecting skyline services for QoS-based web service composition'. *Proceedings of the 19th international conference on World Wide Web (WWW 2010)*, North Carolina USA, pp. 11-20.

Ardagna, D. and Pernici, B. (2007) 'Adaptive Service Composition in Flexible Processes', *Journal of the American Medical Association*, Vol. 144, No. 18, pp. 1540-3.

Barakat, L., Miles, S. and Luck, M. (2012) 'Efficient Correlation-Aware Service Selection'. *Proceedings of the IEEE International Conference on Web Services (ICWS 2012)*, Honolulu, Hawaii, USA, pp. 1-8.

Borzsony, S., Kossmann, D. and Stocker, K. (2001) 'The Skyline operator'. *Proceedings of the 17th International Conference on Data Engineering (ICDE 2001)*, Heidelberg, Germany pp. 421-430.

Canfora, G., Penta, M. D., Esposito, R. and Villani, M. L. (2008) 'A framework for QoS-aware binding and re-binding of composite web services', *Journal of Systems & Software,* Vol. 81, No. 10, pp. 1754-1769.

Dai, G. (2013) 'Key Technology Research on Web Service Selection based on QoS', *Chongqing University*.

Ding, K., Deng, B., Zhang, X. Y. and Ge, L. (2009) 'Optimization of service selection algorithm for complex event processing in Enterprise Service Bus platform'. *Proceedings of the 4th International Conference on Computer Science and Education (ICCSE 2009)*, Nanning, China, pp. 582 - 586.

Feng, Y. Z., Le, D. N. and Kanagasabai, R. (2013) 'Dynamic Service Composition with Service-Dependent QoS Attributes'. *Proceedings of the IEEE International Conference on Web*

Services (ICWS 2013), Santa Clara, California, USA, pp. 10-17.

Gao,Y., Na, J., Zhang, B., Yang, L. and Gong, Q. (2006) 'Optimal Web Services Selection Using Dynamic Programming', *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC 2006)*,Cagliari, Sardinia, Italy, pp. 365-370.

Hiratsuka, N., Ishikawa, F. and Honiden, S. (2011) 'Service Selection with Combinational Use of Functionally-Equivalent Services'. *Proceedings of the IEEE International Conference on Web Services (ICWS 2011)*, Washington, DC, USA, pp. 97-104

Hwang, S. Y., Lim, E. P., Lee, C. H. and Chen, C. H. (2008) 'Dynamic Web Service Selection for Reliable Web Service Composition', *IEEE Transactions on Services Computing*, Vol.1, No.2, pp. 104-116.

Janacek, G. (1990) 'Time series analysis forecasting and control': *Holden-Day, Incorporated.*

Jang, J. h., Shin, D. h. and Lee, K. h. (2006) 'Fast Quality Driven Selection of Composite Web Services'. *Proceedings of the 4th IEEE European Conference on Web Services (ECOWS 2006)*, Zürich, Switzerland. pp. 87-98.

Li, F. and Huang, Y. (2010) 'An web service selection optimization approach based on particle swarm optimization'. *Proceedings of the International Conference on Computer Design and Applications (ICCDA 2010)*, Qinhuangdao, China, pp. V2-477 - V2-481.

Liu, Z. Z., Wang, Z. J., Zhou, X. F., Lou, Y. S. and Shang, L. (2010) "A New Algorithm for QoS-Aware Composite Web Services Selection." *Proceedings of the 2nd International Workshop on Intelligent Systems and Applications (ISA 2010)*, Wuhan, China pp. 1-4.

Lin, W. M., Dou, W. C., L, X. F. and Chen, J. (2011) 'A History Record-Based Service Optimization Approach for QoS-Aware Service Composition'. *Proceedings of the IEEE International Conference on Web Services (ICWS 2011)*, Washington, DC, USA, pp. 666-673.

Ma Y, Wang, S. G., Hung PCK, Hsu C-H, Sun, Q. B. and Yang, F. C. (2016) 'A Highly Accurate Prediction Algorithm for Unknown Web Service QoS Value'. *IEEE Transactions on Services Computing*, Vol. 9, No. 4, pp. 511-523.

Sun, L., Wang, S. G. Li, J. L. and Sun, Q.B. (2014) 'QoS Uncertainty Filtering for Fast and Reliable Web Service Selection'. *Proceedings of the IEEE International Conference on Web Services (ICWS 2014)*, Anchorage, Alaska, USA, pp. 550-557.

Tang, M. L. and Ai, L. F. (2010) 'A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition'. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2010)*, Barcelona, Spain, pp. 1-8.

Wang, S. G. (2011) 'Research on Key Technologies for Web Service Selection based on QoS Measure', *Beijing University of Posts and Telecommunications*.

Wang, S. G., Huang, L., Hsu, C-H. and Yang, F. C. (2016) 'Collaboration reputation for trustworthy Web service selection in social networks'. *Journal of Computer and System Sciences*, Vol. 82, No.1, pp.130-143.

Wang, S. G., Liu, Z. P., Sun, Q. B., Zou, H. and Yang, F. C. (2014) 'Towards an Accurate Evaluation of Quality of Cloud Service in Service-oriented Cloud Computing'. *Journal of Intelligent Manufacturing*, Vol. 25, No. 2, pp. 283-291

Wang, S. G., Sun, Q. B., Zou, H. and Yang, F. C. (2013) 'Particle Swarm Optimization with Skyline Operator for Fast Cloud-based Web Service Composition'. *Mobile Networks and Applications*, Vol. 18, No.1, pp. 116-121.

Wang, S. G., Sun, Q. B., Zhang, G. W. and Yang, F. C. (2012) 'Uncertain QoS-Aware Skyline Service Selection Based on Cloud Model'. *Journal of software* , No.06, pp. 1397-1412.

Wang, S. G., Sun, L., Sun, Q. B., Li, J. L. and Yang, F. C. (2014) 'Efficient Service Selection in Mobile Information Systems', *Mobile Information Systems*, Vol. 10, No. 2, pp. 197-215.

Wang, S. G., Sun, L., Sun, Q. B., Wei, J. and Yang, F. C. (2015) 'Reputation Measurement of Cloud Services based on Unstable Feedback Ratings'. *International Journal of Web and Grid Services.* Vol. 11, No. 4, pp. 362-376.

Wang, S. G., Zheng, Z. B., Wu, Z. P., Lyu, M. and Yang, F. C. (2015) 'Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation System'. *IEEE Transactions on Services Computing*, Vol. 5, No. 8, pp. 755 - 767.

Wang, S. G., Zheng, Z. B., Sun, Q. B., Zou, H. and Yang, F. C. (2011) 'Reliable web service selection via QoS uncertainty computing'. *International Journal of Web and Grid Services*, Vol.7,

No.4, pp.410-426.

Wang, S. G., Zhou, A., Yang, F. C. and Chang, R. (2016) 'Towards Network-Aware Service Composition in the Cloud'. *IEEE Transactions on Cloud Computing*, DOI: 10.1109/TCC.2016.2603504

Wang, S. G., Zhu, X. L., Sun, Q. B., Zou, H. and Yang, F. C. (2013) 'Low-Cost Web Service Discovery Based on Distributed Decision Tree in P2P Environments'. *Wireless Personal Communications*,Vol. 73, Issue 4,pp 1477-1493.

Wan, C. L., Ullrich, C., Chen, L. M., Huang, R., Luo, J. M. and Shi, Z.Z. (2008) 'On Solving QoS-Aware Service Selection Problem with Service Composition'. *Proceedings of the 7th International Conference on Grid and Cooperative Computing (GCC 2008)*, Shenzhen, China, pp. 467-474.

Zheng, Z., Zhang, Y. and Lyu, M. R. (2010) 'Distributed QoS Evaluation for Real-World Web Services'. *Proceedings of the IEEE International Conference on Web Services (ICWS 2010)*, Miami, USA, pp. 83-90.