

## Research Article

# A Robust Service Selection Method Based on Uncertain QoS

Yanping Chen,<sup>1</sup> Lu Jiang,<sup>2</sup> Jianke Zhang,<sup>3</sup> and Xiaoxiao Dong<sup>1</sup>

<sup>1</sup>School of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

<sup>2</sup>ZTE Corporation, Xi'an 710114, China

<sup>3</sup>School of Science, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Correspondence should be addressed to Yanping Chen; [chenyp@xupt.edu.cn](mailto:chenyp@xupt.edu.cn)

Received 14 October 2015; Revised 11 January 2016; Accepted 27 January 2016

Academic Editor: Alireza Amirteimoori

Copyright © 2016 Yanping Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, the number of Web services on the Internet is quickly increasing. Meanwhile, different service providers offer numerous services with the similar functions. Quality of Service (QoS) has become an important factor used to select the most appropriate service for users. The most prominent QoS-based service selection models only take the certain attributes into account, which is an ideal assumption. In the real world, there are a large number of uncertain factors. In particular, at the runtime, QoS may become very poor or unacceptable. In order to solve the problem, a global service selection model based on uncertain QoS was proposed, including the corresponding normalization and aggregation functions, and then a robust optimization model adopted to transform the model. Experiment results show that the proposed method can effectively select services with high robustness and optimality.

## 1. Introduction

With the wide use of the Web service in various fields, more and more services have been developed rapidly. But single service can only implement single function; service composition becomes the primary measure to provide user with a high-quality and personalized complex service. And the Web service selection is the foundation and precondition of service composition. With the number of Web services increasing gradually, a key issue arises immediately; there are many services providing same functions but differing in the Quality of Service (QoS). Therefore, in the process of service composition, we need to select service, not only depending on the functions but also according to the user's QoS requirement.

Web service selection based on QoS is a combinatorial optimization problem. Traditional QoS-based Web service selection algorithms mainly use linear programming [1] and Greedy Algorithm [2]. Currently more popular service selection algorithms are mainly using smart algorithms, such as Genetic Algorithms [3], Particle Swarm Optimization [4], and Ant Colony Optimization Algorithm [5]. These algorithms are often designed for certain QoS attributes and assumed that, after the service provider has provided services, the services would run as the described state without change.

But in fact, this ideal situation does not exist. Web service is provided as a distributed, loosely coupling API on the Web, so QoS is uncertain factor; for example, the response time would fluctuate with the change of the network. It will fluctuate the QoS of the composed services which are solved by the deterministic algorithms at the running time and cannot satisfy the QoS requirements of users or is unusable. Therefore, when dealing with the service selection problem, the uncertain factors must be considered.

Uncertainty is the most important factor that the requirements of users cannot be satisfied. We consider that the uncertainty of the service composition is mainly caused by the following reasons:

- (1) QoS information is incomplete. The Web service description emphasizes more on functional information, so the description of QoS attributes is not accurate enough.
- (2) There is no unified QoS metric of Web services, and only one way can gain service QoS value from service providers. Due to the fuzziness of the natural language, there may be divergence between the service providers and the users even about the same attributes. For example, there is one attribute called

throughput, one definition is the number of requests for a service successfully executed, and the other is the average rate of successful message size delivery over a communication channel per second.

- (3) Internet environment cannot be repeated. When services are used on the Internet, some attributes of QoS of the Web services are influenced by the running state of the Internet, such as security and response time. The real situation of service cannot be described by the QoS obtained from once request.

Because of the existing uncertainties, the composite service could hardly be accomplished in the dynamic execution process. These uncertainties also affect the individual requirements of users. When the QoS deviation occurs, we must replan and compose services; otherwise it would cause a lot of problems. But the replanning will lead to the waste of computing resources, so this paper studies how to ensure that service composition is feasible with the uncertain conditions, thus avoiding the replanning of composited service. For the uncertainty of the QoS properties, this paper combined the uncertain optimization theory, described the uncertainty by using the “interval,” modeled a service composition model based on workflow (order, circulation, choice, and parallel), proposed the service selection model under the conditions with uncertain QoS, put forward the corresponding normalization and aggregation function, and ensured that when the change occurred in the part of QoS attributes, the obtained solution can adapt to the most cases and close to the optimal.

The rest of the sections are organized as follows. Section 2 describes related works. Section 3 details the robust optimization method used in service selection model based on the uncertain QoS. Section 4 presents the experiment and analysis result. Finally, Section 5 draws some conclusions.

## 2. Related Works

In Zeng et al.'s paper [1] QoS was introduced into Web service at the first time; the QoS of Web service was used to distinguish the Web services with similar functions. From then on, service selection based on QoS plays a more and more important role in the Web service composition area. QoS-based service selection methods are mainly divided into two directions: semantic QoS [6] and QoS value. According to the different QoS requirements, the Web service selection strategy can be divided into two kinds, namely, the local optimal strategy and the global optimal strategy. Local optimal strategy selects component services for a single service function, respectively, compares the candidate atomic service set, sorts QoS values for each candidate atomic service, and then selects the best one. The main algorithm is the Greedy Algorithm [2]. Global optimal strategy aims to make the global QoS of the composite services to satisfy the given constraints or achieve the desired goals; the main methods are Genetic Algorithm [3], PSO [4], and Ant Colony Algorithm [5].

In recent years, the researchers have also addressed some studies for service selection based on uncertain QoS. Zeng et al. [1] proposed QoS-based service selection; at the same time they proposed that uncertain response time was introduced

into the service selection model. They assumed the response time of service subjected to normal distribution and introduced the variance of the normal distribution into the constraints. Alférez and Pelechano [7] leveraged models at runtime to guide the dynamic evolution of context-aware Web service compositions to deal with unexpected events in the open world. In order to manage uncertainty, a model that abstracts the Web service composition self-evolves to preserve requirements. Benouaret et al. [8] represented each QoS attribute of a Web service using a possibility distribution and introduced two skyline extensions on uncertain QoS called pos-dominant skyline and nec-dominant skyline. Wu et al. [9] proposed an “on-demand” strategy for QoS-aware service composition to replace the traditional “best-effort” strategy. Mostafa and Zhang [10] presented two approaches based on the Multiobjective Reinforcement Learning (MORL) to address multiobjective service composition in uncertain and dynamic environments. Wagner et al. [11] developed a new QoS model that helps to predict the resulting QoS of a workflow by considering service failures during the initial selection phase. For each service, a set of possible backup services and the corresponding QoS are computed beforehand.

In recent years, it has been recognized that dealing with uncertain data is a major challenge in optimization. As can be seen from the above literatures, Web service QoS models with uncertainty dispose uncertain data in two ways; one would assume that the data is subject to one kind of distributions to compute the uncertainty of QoS and the other solves the problem by using intelligent algorithms. But the problem of the former method is that the assumption may be not reasonable. In the real world, we could not get the exact distribution of the data. And for the latter method, different intelligent algorithms have different problem, like premature convergence for Genetic Algorithm. And for all intelligent algorithms, they always solve different solutions in different times, even if they use the same data. We need an approach to select service considering all possible situations, but it does not take whether the uncertain data is subject to any distributions into account.

The researches of the traditional optimization problems are mostly about deterministic problems. When solving uncertain optimization problems, researchers mainly focus on the following ways: stochastic programming, fuzzy programming, interval programming, and robust optimization. When solving the uncertain problems, the stochastic programming and fuzzy programming need precise distribution of the parameters, and the precise distribution is difficult to obtain in the real world. When solving these two kinds of programming problems, the commonly used method is the Monte Carlo method, but this method shows slow convergence and low efficiency. Facing more complex constraints, they are only to use approximate method. Interval programming and robust optimization are all converting the uncertain optimization to determine problem for solving. The difference is that the robust optimization considers the worst case; the result may be conservative. And the interval programming eventually provides decision-makers with a range of optimal value; they will have a better space to select [12]. When converting the interval programming, upper and lower

bounds of uncertainty goals and constraints will become a nested optimization model, so the problem is always a multi-objective optimization problem and difficult to solve.

In the 1973s, Soyster [13] proposed an approach to address data uncertainty named robust optimization. Robust optimization means finding a solution which can deal with all possible realizations of the uncertain data. The first robust optimization method proposed by Soyster considers the maximum uncertainty. That means only considering the worst case, the uncertain programming is converted into the corresponding deterministic model to obtain robust optimal solution. The model only considers the worst case and leads to the very conservative solutions. Conservatism in robust optimization means that the solution ensures a high robustness level but is “far away” from optimality. In other words, a robust solution is not optimal for all realizations of the uncertain data but performs well even for the worst case. Later researches focus on how to keep the optimality and robustness at the same time. At the end of the last century, Ben-Tal and Nemirovski [14] gave an approach that converted the linear programming to the robust counterpart problem with the ellipsoid uncertain set, but the converted problem is a Second-Order Cone Programming (SOCP) and difficult to calculate. In 2003, Bertsimas and Sim [15, 16] proposed a new robust approach which obtained the solution considering optimality and robustness and analyzed the probability bounds of constraint violations. Existing researches based on the two approaches used the robust optimization in the field of supply chain planning, network flow, and logistics programming.

In this paper, we propose using robust optimization method to select service. The difference between the existing study and this method is that this method considers all possible situations during the service as invoking. And it is solved by mathematical methods, which means that if the data is constant, the results are constant.

### 3. Service Selection Method Based on Robust Optimization

*3.1. QoS of the Candidate Services.* Since Zeng et al. [1] combined Web service with QoS together, there have been a large number of QoS attribute definitions which describe nonfunctional properties of Web services. In many literatures, QoS attributes are using the determinate data to make it clear, but some QoS attributes should be described as uncertain interval data, such as response time. This paper uses the following three kinds of common QoS attributes as a basis for service selection. Defining that response time and throughput are described by the interval data and the reliability is real data. We consider that the value of QoS is not subject to any distribution.

*Definition 1 (response time).* Response time measures the expected delay in seconds from a request sent to the results which are received [1]. It is computed using the expression as follows:

$$Q_{rt} = T_{tran} + T_{proc}, \quad (1)$$

where  $T_{tran}$  is the transmission time of request and response in the network.  $T_{proc}$  is the time of processing the request.

In the past literatures, the response time is often using the minimum value or average value of the response time in a period of time, as a certain value to calculate. But the nature of the response time should be uncertain. Due to the different network status, the same service will show different response times. Using traditional QoS models always masks the uncertainty of response time, ignoring the unstable factors. Without considering the uncertainty of response time, the selected service would take a bad response time. For example, the minimum and average response time of service A, respectively, are 0.1 s and 0.2 s and of service B are 0.15 s and 0.22 s. However, the intervals of response time of services A and B, respectively, are [0.1, 1.0] and [0.15, 0.5]. From the two intervals, it can be seen that the response time of service B is more stable and robust. Using the traditional service selection method, service A is better than service B. When we use the two services, the response time of service A may be 0.9 s and that of service B may be 0.3. The purpose of this paper is to choose a more robust service. We use interval  $[Q_{rt}, Q_{rt} + \Delta Q_{rt}]$  to indicate the uncertainty of response time.

*Definition 2 (throughput).* For the throughput, one definition is the number of requests for a service successfully executed, and the other is the average rate of successful message size delivery over a communication channel per second. In this paper, we define it as in the second definition. It is also uncertain. So we describe the throughput like response time with  $[Q_{tp}, Q_{tp} + \Delta Q_{tp}]$ .

*Definition 3 (reliability).* Reliability of service is the probability of a request that is correctly responded to within the maximum number of total requests. It is computed using the expression as follows:

$$Q_{re} = \frac{N_{re}}{n}, \quad (2)$$

where  $N_{re}$  means the number of successful requests in  $n$  invocations and  $n$  is the total number of service executions. In this paper, the reliability is derived from the service attribute dataset and is a certain value.

*3.2. Service Composition Based on Workflow.* Most of single Web service is usually providing only a single way to implement a single function, but the composite services can take advantage of many small particle size distributions of the atomic services on the network and combine them to be able to implement complex functions, loosely coupled composite services.

As shown in Figure 1, business logic of each kind of complex application is fixed, which has been divided into service classes by domain experts, and then the service class is advertised. After that, service providers could publish their Web service into registry and point out which certain service class it belongs to. When users' requirement is coming (meaning that user selects a certain business logic), then the proper services should be selected according to users' QoS. In order to solve the problem, a global service selection model based

TABLE I: Aggregate formulas of QoS.

	Response time	Throughput	Reliability
Sequence	$\left[ \sum_{i=1}^n Q_{rti}, \sum_{i=1}^n (Q_{rti} + \Delta Q_{rti}) \right]$	$\left[ \min_i Q_{tpi}, \min_i (Q_{tpi} + \Delta Q_{tpi}) \right]$	$\prod_{i=1}^n Q_{rei}$
Selection	$\left[ \sum_{i=1}^n c_i Q_{rti}, \sum_{i=1}^n c_i (Q_{rti} + \Delta Q_{rti}) \right]$	$\left[ \min_i c_i Q_{tpi}, \min_i c_i (Q_{tpi} + \Delta Q_{tpi}) \right]$	$\sum_{i=1}^n c_i Q_{rei}$
Loops	$[kQ_{rt}, k(Q_{rt} + \Delta Q_{rt})]$	$[Q_{tp}, Q_{tp} + \Delta Q_{tp}]$	$Q_{re}^k$
Parallel	$\left[ \max_i Q_{rti}, \max_i (Q_{rti} + \Delta Q_{rti}) \right]$	$\left[ \min_i Q_{tpi}, \min_i (Q_{tpi} + \Delta Q_{tpi}) \right]$	$\prod_{i=1}^n Q_{rei}$

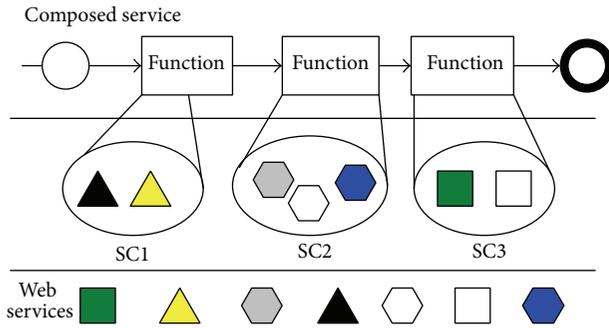


FIGURE 1: QoS-based Web services composition problem.

on uncertain QoS was proposed, including the corresponding normalization and aggregation functions, and then a robust optimization model is adopted to transform the model.

Techniques of composite services are various. Mainly methods include modeling based on workflow, modeling based on Petri nets, and semantic modeling. This paper focuses on how to select the atomic service to accomplish the function, regardless of the composite modeling techniques; how to choose the specific combination service workflow model is also the key problem.

Workflow model has four basic atomic structures, namely, sequence, selection, loop, and parallel. The flow of composite services can be combined or nested by the basic structures. If we can find a corresponding calculation method of each QoS for each basic structure, we can give the global QoS of composite services. In the traditional QoS-driven service selection algorithms when aggregating the QoS of the services, the algorithms only consider the determined QoS, so that the aggregate formula is simple. But there is uncertainty in the QoS, and we described it by interval numbers, so the traditional function cannot be used for calculating global QoS. In order to solve this problem, we combined the interval number algorithm with the characteristics of response time and throughput and proposed the aggregation formulas of three QoS properties corresponding to each basic structure in Table 1.

**3.3. Normalization.** Different QoS attributes always have different dimensions. Some value of QoS attributes is the bigger the better, such as throughput and reliability, and this kind of attributes is called benefit type attribute; some is the smaller

the better, such as response time, called cost type attribute. In this case, integrating global QoS of composite service could result in additional cost. In order to easily calculate and compare, data in the decision matrix needs to be normalized into the same dimension. The service selection algorithm based on real number generally used 0-1 normalization method; the attributes' values are converted to interval  $[0, 1]$  by linear calculations. However, in this paper, response time and throughput are interval data and cannot use the method of real number to normalize. The researches about normalization methods of interval data are not too many and mainly focused on two ways. One is to convert the interval data into a real number. The other is to use some kinds of normalization functions to directly convert the interval number; the result is still interval number. While the former method will lose the uncertain information which is what this paper focused on, we use the latter method.

Common methods convert interval numbers into standardized interval number including the method based on interval number operation method, vector-based normalization method, method based on a linear scale transformation, and method based on the range transformation [17]. These three former methods combine the normalization method of real number with interval number operation method. However, the data obtained by these methods always is out of  $[0, 1]$ , and these methods are based on nonlinear transformation. There has been a huge impact on the decision results. So we use the method based on the range transformation to normalize response time and throughput. Because the value of the reliability is taken in the range of  $[0, 1]$ , it does not need to be normalized.

There are  $n$  services realizing one function. Assuming that the QoS attribute of each service takes values in the interval  $[Q_i, Q_i + \Delta Q_i]$ , the QoS after normalization is written as  $[Q'_i, Q'_i + \Delta Q'_i]$ . In order to easily calculate, we need to normalize the attributes data to be cost type data. The normalization formula is established as follows:

Cost type:

$$Q'_i = \frac{Q_i - \min_i Q_i}{\max_i (Q_i + \Delta Q_i) - \min_i Q_i}, \quad (3)$$

$$Q'_i + \Delta Q'_i = \frac{(Q_i + \Delta Q_i) - \min_i Q_i}{\max_i (Q_i + \Delta Q_i) - \min_i Q_i}.$$

Benefit type:

$$\begin{aligned} Q'_i &= \frac{\max_i(Q_i + \Delta Q_i) - Q_i}{\max_i(Q_i + \Delta Q_i) - \min_i Q_i}, \\ Q'_i + \Delta Q'_i &= \frac{\max_i(Q_i + \Delta Q_i) - (Q_i + \Delta Q_i)}{\max_i(Q_i + \Delta Q_i) - \min_i Q_i}. \end{aligned} \quad (4)$$

**3.4. Service Selection Model Based on Uncertain QoS.** At present, many scholars transform the service selection problem as mathematical problems from different ways and designed the corresponding selection model. The transformed mathematical problems include linear or integer programming problem, multiple choice knapsack problem, multiple attribute decision-making problems, single objective combinatorial optimization problems with QoS constraints, and multiobjective combinatorial optimization problem with QoS constrained. But these problems always do not consider the uncertainty of QoS, so we propose a service selection model considering the uncertainty.

According to the global service selection strategy, we propose a service selection model based on uncertain QoS. Assuming that there are  $p$  tasks in the workflow, each task has  $q$  candidate atomic services, so that the number of all possible composite services is  $q^p$ . We propose a method based on robust optimization (RO), which can select one service with high robustness level from  $q^p$  composite services. In order to implement our approach, firstly we need to formulate a linear programming model.

There are three inputs in linear programming problem: variables, objective function, and constraint conditions, in which the objective function and constraints must be linear. Linear programming is to maximize or minimize the value of objective function by adjusting the value of the variables. The output of linear programming is maximum (or minimum) value of objective function. We assume that the variable is  $x_{ij}$ , which means whether the service  $s_{ij}$  is selected to execute plan or not. If the service  $s_{ij}$  is selected, then the value of the variable  $x_{ij}$  is 1; otherwise it is 0. The objective function of the model is the following formula:

$$\min Q_{rt} \cdot \omega_1 + Q_{tp} \cdot \omega_2 + Q_{re} \cdot \omega_3, \quad (5)$$

where we transformed three QoS attributes into cost type attributes and  $Q_{rt}$ ,  $Q_{tp}$ , and  $Q_{re}$  are global QoS.  $\omega_i$  is the weight of the QoS, and  $\sum_{i=1}^3 \omega_i = 1$ ,  $\omega_i \in [0, 1]$ .

We used  $Q_{rt}(x_{ij})$  to express the response time of service  $s_{ij}$  for task  $t_i$ , so we have the following constraint:

$$Q_{rt} = \sum_{i=1}^n \sum_{j=1}^{m_i} Q_{rt}(x_{ij}) \cdot x_{ij}. \quad (6)$$

Assume that there are  $n$  workflow tasks and there is a set of  $m_i$  services that can be provided to the user for each task, but there is only one service selected in each set, so  $x_{ij}$  ( $x_{ij} = 0, 1$ ) indicates whether the service is selected.  $x_{ij}$  has the following constraints:

$$\sum_{j=1}^{m_i} x_{ij} = 1. \quad (7)$$

For example, there are 100 candidate Web services that can execute the task  $t_i$ ; however, only one of them will be selected to the task, so it is  $\sum_{j=1}^{100} x_{ij} = 1$ .

We used  $Q_{tp}(x_{ij})$  to express the throughput of service  $s_{ij}$ . From Table 1, we can see that the aggregate formula of throughput is looking for the minimum throughput in the process as the global throughput. But before the calculation of QoS value, we need to normalize it; the throughput will be converted from the benefit type to the cost type, so the global throughput is the maximum value. The constraint to the throughput is as follows:

$$Q_{tp} = \max Q_{tp}(x_{ij}) \cdot x_{ij}. \quad (8)$$

$Q_{re}(x_{ij})$  means the reliability of service  $s_{ij}$ . Like throughput, the reliability also needs to be converted, so the global reliability is as follows:

$$Q_{re} = 1 - \prod_{i=1}^n \prod_{j=1}^{m_i} Q_{re}(x_{ij}) \cdot x_{ij}. \quad (9)$$

However, the constraint of attribute can be introduced into the linear programming problem which is not limited to the defined one in this section. If the aggregate function could be given, it also can be added to the model.

We can propose the following model:

$$\begin{aligned} \min \quad & \left\{ \omega_1 \sum_{i=1}^n \sum_{j=1}^{m_i} Q_{rt}(x_{ij}) \cdot x_{ij} + \omega_2 \cdot \max Q_{tp}(x_{ij}) \cdot x_{ij} + \omega_3 \left( 1 - \prod_{i=1}^n \prod_{j=1}^{m_i} Q_{re}(x_{ij}) \cdot x_{ij} \right) \right\} \\ \text{s.t.} \quad & \sum_{i=1}^3 \omega_i = 1 \\ & \sum_{i=1}^n x_i = 1 \\ & x_{ij} \in \{0, 1\}, \end{aligned} \quad (10)$$

where there exists uncertainty in two QoS attributes: response time and throughput. For each entry  $Q_{rt}(x_{ij})$  is independent and bounded random variable and takes values in  $[Q_{rt}(x_{ij}), Q_{rt}(x_{ij}) + \Delta Q_{rt}(x_{ij})]$ .  $\Delta Q_{rt}(x_{ij})$  means the deviation of the coefficient  $Q_{rt}(x_{ij})$ . It is the same as throughput, in that, for each entry,  $Q_{tp}(x_{ij})$  is independent and bounded random variable and takes values in  $[Q_{tp}(x_{ij}), Q_{tp}(x_{ij}) + \Delta Q_{tp}(x_{ij})]$  and  $\Delta Q_{tp}(x_{ij})$  means the deviation of the coefficient  $Q_{tp}(x_{ij})$ .

$$\begin{aligned} \min \quad & \left\{ \omega_1 \sum_{i=1}^n \sum_{j=1}^{m_i} Q_{rt}(x_{ij}) \cdot x_{ij} + \omega_2 \cdot \max Q_{tp}(x_{ij}) \cdot x_{ij} + \omega_3 \left( 1 - \prod_{i=1}^n \prod_{j=1}^{m_i} Q_{re}(x_{ij}) \cdot x_{ij} \right) + \max_{\{S|S \subseteq A, |S| \leq \Gamma\}} \left\{ \omega_1 \sum_{(i,j) \in S} \Delta Q_{rt}(x_{ij}) \cdot x_{ij} + \omega_2 \max \Delta Q_{tp}(x_{ij}) \cdot x_{ij} \right\} \right\} \\ \text{s.t.} \quad & \sum_{i=1}^n \omega_i = 1, \quad \omega_i \in [0, 1] \\ & \sum_{j=1}^{m_i} x_{ij} = 1 \\ & x_{ij} \in \{0, 1\}. \end{aligned} \quad (11)$$

$\Gamma$  in objective function (11) is to adjust the robustness of the solution against the level of conservatism of the solution.  $\Gamma = 0$ : the model will completely ignore all uncertain factors, so that the solution is the same as the solution obtained by solving with minimum values  $Q_{rt}(x_{ij})$  and  $Q_{tp}(x_{ij})$ ; this solution is called the optimal solution.  $\Gamma = 1$ : the model will consider all uncertainties, but the solution is too conservative; this solution is called robust solution. If the value of  $\Gamma$  is more close to 1, we say the corresponding solution has a higher level of robustness. So robust solution has the highest level of robustness. In general, a higher value of  $\Gamma$  increases the level of robustness at the expense of higher nominal cost.

#### 4. Experiment Analyses

In order to verify the effectiveness of the proposed method, we developed Web service testing and selection management system in the early study. In the literature, the researchers usually solved deterministic model. The value of QoS they used is from one test result or some random numbers produced in a certain range. This paper is to study and calculate uncertain data, once test result or random numbers cannot describe uncertainty of the service used in the real world. We use the response time and throughput datasets from Zheng et al.'s paper. In the literature [18], about 1.9 million pieces of data were provided by 339 users calling the 5825 atomic services. In this dataset, we found that when users invoked the service, the request may succeed or not, and the values of response time and throughput are always varied.

In this experiment, the workflow is a simple sequence flow with four tasks; each task would invoke in order. Each task has 10 candidate services, so the number of all possible composite services is  $10^4$ . Set three QoS attributes that have the same weight and calculate the uncertainty QoS interval of

Obviously, model (10) cannot be solved with the traditional optimization method or intelligent algorithm. In order to be able to consider the uncertainty of QoS in the solution, we used the robust optimization model proposed by Bertsimas to deal with the uncertainty. Due to the uncertainty that only appears in the objective function, we introduce a parameter  $\Gamma$ , which takes value in the interval  $[0, k]$ , where  $k$  is the number of selected services in the model. For one flow, there will be only one composite service selected, so the range of  $\Gamma$  should be  $[0, 1]$ . So we propose the following service selection model:

composite service through the front of the aggregation function. Table 2 is the QoS value of 40 candidate services.

In Table 2, the data comes from Zheng et al.'s dataset. We find out the maximum and minimum value as the upper and lower bounds of the response time and throughput intervals. In the dataset,  $-1$  means the request is failed, so we calculate the reliability by failure number and total number.

Since  $\Gamma > 1$ , the obtained composite service is the same as when  $\Gamma = 1$ , so in the experiment,  $\Gamma$  takes value in the interval  $[0, 1]$ . If  $\Gamma$  is an integer, it can only take 0 or 1; in both cases, obtained composite services are, respectively, the optimal solution and the robust solution mentioned in Section 3.2. To validate the method,  $\Gamma$  will be taking decimal to analysis.

When solving (11), we have two ways. The first way is solving the robust counterpart of (11), and the second is using one algorithm. The solving algorithm is as follows:

- (1) For  $l = 1, \dots, n + 1$ , solve the  $n + 1$  nominal problems

$$G^l = \Gamma \cdot d_l + \min_{x \in X} \left( c' \cdot x + \sum_{j=1}^l (d_j - d_l) \cdot x_j \right) \quad (12)$$

and let  $l^*$  be an optimal solution of the corresponding problem.

- (2) Let  $l^* = \arg \min_{l=1, \dots, n+1} G^l$ .
- (3)  $Z^* = G^{l^*}$ ,  $\mathbf{x}^* = \mathbf{x}^{l^*}$ .

Although there are a lot of methods and tools for solving integer programming, it would spend too much time, and the algorithm can quickly find a solution in few seconds. So this experiment uses the above algorithm to solve it.

The experiment is divided into two parts. The first part is to adjust the value of  $\Gamma$  and observe parameter effect on

TABLE 2: The QoS value of 40 candidate services.

Number	$Q_{rt}$	$Q_{tp}$	$Q_{re}$
Task 1			
1	[0.432, 16.523]	[0.02, 4.629]	97.94%
2	[0.078, 4.29]	[0.932, 51.282]	98.23%
3	[0.084, 13.835]	[0.433, 71.428]	98.53%
4	[0.061, 3.317]	[0.602, 32.786]	98.53%
5	[0.071, 3.103]	[0.644, 28.169]	98.53%
6	[0.015, 6.582]	[0.455, 200.0]	99.71%
7	[0.154, 9.626]	[0.207, 12.987]	99.71%
8	[0.22, 9.552]	[1.201, 486.363]	99.71%
9	[0.115, 3.726]	[0.536, 17.391]	98.53%
10	[0.169, 13.237]	[0.226, 17.751]	98.23%
Task 2			
11	[0.301, 16.648]	[0.12, 6.644]	97.94%
12	[0.444, 15.867]	[0.126, 4.504]	94.69%
13	[1.541, 16.807]	[3.093, 33.744]	98.82%
14	[0.455, 9.651]	[1.347, 28.571]	97.94%
15	[0.191, 19.652]	[0.101, 10.471]	74.63%
16	[0.116, 15.877]	[0.06, 17.241]	98.23%
17	[0.466, 11.27]	[0.798, 19.313]	98.23%
18	[0.502, 11.187]	[0.804, 17.928]	97.94%
19	[0.284, 10.87]	[0.275, 10.563]	100.00%
20	[0.164, 2.149]	[1.395, 18.292]	100.00%
Task 3			
21	[0.164, 3.594]	[0.834, 18.292]	100.00%
22	[0.325, 9.772]	[0.306, 9.23]	100.00%
23	[0.162, 3.512]	[0.854, 18.518]	100.00%
24	[0.162, 3.816]	[0.786, 18.518]	100.00%
25	[0.345, 7.115]	[0.421, 8.695]	100.00%
26	[0.17, 2.066]	[1.452, 17.647]	100.00%
27	[0.169, 2.173]	[1.38, 17.751]	100.00%
28	[0.17, 2.349]	[1.277, 17.647]	100.00%
29	[0.379, 14.363]	[0.208, 7.915]	100.00%
30	[0.194, 4.32]	[0.694, 15.463]	100.00%
Task 4			
31	[0.195, 14.418]	[0.208, 15.384]	100.00%
32	[0.217, 13.192]	[0.227, 13.824]	98.82%
33	[0.2, 3.272]	[0.916, 15.0]	98.82%
34	[0.202, 3.61]	[0.831, 14.851]	98.82%
35	[0.383, 19.796]	[0.146, 7.832]	99.41%
36	[0.216, 5.397]	[0.555, 13.888]	99.71%
37	[0.276, 11.531]	[0.433, 18.115]	98.53%
38	[0.496, 16.851]	[0.534, 18.145]	100.00%
39	[0.422, 15.991]	[2.814, 106.635]	100.00%
40	[0.343, 9.15]	[0.546, 14.577]	100.00%

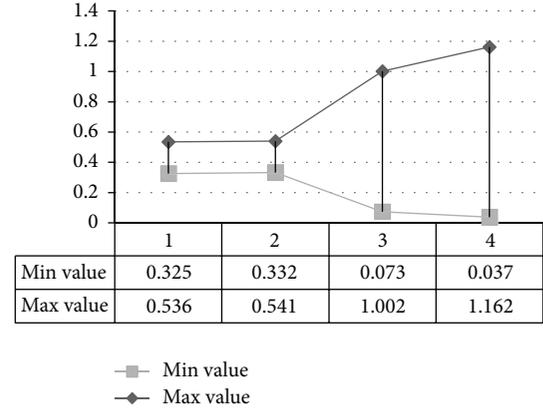


FIGURE 2: The global QoS range of four composite services.

In order to compare the results of the different methods, this paper will show the global QoS of services which are selected by methods. If the global QoS of the selected composite service is lower, we believe that the optimality of the service is better. If the deviation of service QoS is smaller and the upper QoS is smaller, we believe the robustness of the service is better. In all results of the experiment, the QoS values of all component services are standardized before, except that the total QoS in Table 4 is the result of the standardization and the aggregation.

4.1. *Effect of Adjusting the Value of  $\Gamma$ .* In the experiment, the value of  $\Gamma$  only can be taken in the interval  $[0, 1]$ , so choose  $\Gamma \in \{1, 0.8, 0.6, 0.4, 0.2, 0\}$  to solve the proposed model. The specific results are shown in Table 3.

It can be seen from Table 3 that when  $\Gamma$  dropped to 0.4, the atomic services of composite service began to change. In Table 4, we give out the QoS attributes of four composite services.

The global QoS has been normalized and aggregated. When we are calculating the global QoS, first we adjust reliability to cost type attribute (the smaller the better) and then sum the three QoS attributes with weight (weight of three QoS attributes is equal; i.e.,  $\omega_1 = \omega_2 = \omega_3 = 1/3$ ).

We named the four combination services from  $CS_1$  to  $CS_4$ . It can be seen from Figure 2 that  $\Gamma$  takes value from 1 to 0, and the deviations of global QoS are increasing. If the services are invoked in the real world, we contrast  $CS_4$  and  $CS_1$ ; the former may realize a smaller QoS value, but the latter is more stable and more robust.

The experiment proved that we can take a tradeoff between optimality and robustness by adjusting the value of  $\Gamma$  with user demand.

4.2. *Contrast RO, LP, and LPURT.* First, respectively, we use three methods for selecting services in the mentioned workflow, obtaining three services called composite services A, B, and C, as shown in Table 5. Among them, when using LP, the response time is the minimum response time of each atomic service, and throughput is the maximum throughput (the best case). As noted above, all of these methods consider

the results by solving the robust optimization method. The second part is to analyze the different results solved by using robust optimization (RO), linear programming (LP), and Linear Programming with Uncertain Response Time [1] (LPURT).

TABLE 3: Solutions of different values of  $\Gamma$ .

Values of $\Gamma$	1	0.8	0.6	0.4	0.2	0
Result of service selection	[5, 20, 26, 33]	[5, 20, 26, 33]	[5, 20, 26, 33]	[4, 20, 26, 33]	[8, 14, 26, 39]	[8, 13, 24, 39]

TABLE 4: The QoS attributes of four composite services.

Number	$\Gamma$	Composite service	Response time	Throughput	Reliability	Global QoS
1	1	[5, 20, 26, 33]	[0.605, 10.59]	[1.452, 28.169]	97.30%	[0.325, 0.536]
2	0.4	[4, 20, 26, 33]	[0.595, 10.804]	[1.452, 32.786]	97.30%	[0.332, 0.541]
3	0.2	[8, 14, 26, 39]	[1.267, 37.26]	[11.201, 486.36]	97.60%	[0.073, 1.002]
4	0	[8, 13, 24, 39]	[2.345, 46.166]	[11.201, 486.36]	98.50%	[0.037, 1.162]

TABLE 5: The results of three methods.

Solving method	Robust optimization	Linear programming	Linear programming with uncertain response time
Composite service	A	B	C
Atomic services	[5, 20, 26, 33]	[6, 16, 23, 31]	[4, 20, 24, 33]
Response time	[0.605, 10.59]	[0.488, 40.389]	[0.587, 12.554]
Throughput	[0.644, 15]	[0.06, 15.384]	[0.602, 15]
Reliability	97.32%	97.91%	97.32%

TABLE 6: The global QoS of three selected composite services.

Methods	Composite service (CS)	Atomic services	User ID	1	2	3	4	5	6	7	8	9	10
RO	A	[5, 20, 26, 33]	Response time	1.51	1.517	0.956	0.92	0.989	1.361	1.582	1.001	1.469	1.004
			Throughput	4.878	6.369	9.049	8.13	8.583	6.172	4.538	8.474	6.355	7.272
LP	B	[6, 16, 23, 31]	Response time	4.011	2.727	1.268	1.109	1.677	2.386	2.053	1.649	2.355	1.074
			Throughput	1.066	1.809	5.181	4.672	2.702	1.915	2.929	2.832	2.057	5.665
URTLP	C	[4, 20, 24, 33]	Response time	1.536	1.525	0.959	0.94	0.993	4.504	1.527	1.009	1.467	1.013
			Throughput	4.962	6.153	9.09	8.032	8.474	0.614	4.538	8.51	6.355	7.194

determined values. When using LPURT, the response time is assumed to be subject to a normal distribution in [1], while throughput still is the maximum throughput without uncertainty.

As it is shown in Table 5, there are two different atomic services between composite services A and C. They have the same reliability, but the deviations of response time and throughput of composite service A are smaller. In the best case, composite service B is seen a little better than A and C; however, the deviation is bigger than the other ones.

Then, we select randomly QoS of 40 candidate atomic services invoked by 10 users from dataset and compute global QoS of composite services. Table 6 and Figure 2 show the global QoS of three selected composite services.

We firstly compare composite services A and B. From the data in Figure 3 and Table 6, we can see that response time and throughput of composite service A are obviously better than B. Then compare composite services A and C. From the data, the QoS of composite service A is better than service C, in most cases. From Figure 3, we can see the QoS of the sixth user and response time and throughput of composite service C have an obviously deteriorated case. After checking the QoS

TABLE 7: The QoS fluctuation of combination services.

	Deviation of response time	Deviation of throughput
Composite service A	0.662	4.511
Composite service B	2.937	4.599
Composite service C	3.564	8.476

of each atomic service, we found that one atomic service of composite service C has a bad case at this user. The reason for this is that LPURT assumed the response time of atomic service subject to normal distribution and introduced the mean and variance to the constraints. While, in the real world, we cannot confirm what the probability distribution of the response time of services is subject to, when we request some services, their quality may be more likely to deteriorate as we have mentioned. Here we could calculate the QoS deviations of the three composite services by 10 users invoked, as shown in Table 7. From Table 7, we can see that the response time and throughput deviations of composite service A are minimal. Influenced by the result of the sixth user, the deviation

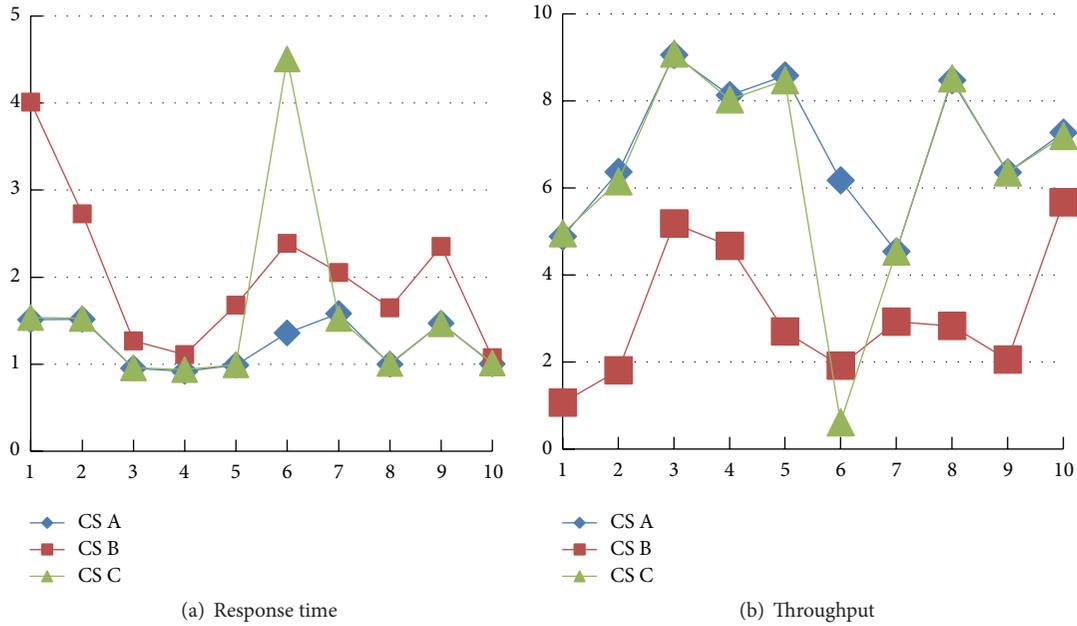


FIGURE 3: The global QoS of three selected composite services.

of QoS of composite service C is maximum. That means robustness level of service A is higher than B and C.

Therefore, the composite service selected by our approach has optimality and robustness. And we can control the tradeoff between optimality and robustness by adjusting the attribute. The proposed method without considering whether the uncertain QoS is subject to some distributions is more suitable for the real network environment.

### 5. Conclusions

This paper studies the robust optimization used in the Web service selection based on uncertain QoS. On the basis of certain QoS-based Web service selection method, first we considered the uncertainty of QoS, second established uncertain QoS-based service selection model, and then used robust optimization methods to convert and solve the proposed model. In the end, the experiment verifies that the proposed model and method can deal with uncertain data without obvious distribution. And we compared the composite services obtained by our approach and other two methods; the result showed that the service selected by our approach had higher robustness and optimality levels.

Uncertain QoS-based Web service composition has the following advantages: firstly, for the software developer, paying attention to this can improve users' acceptance, enhance product competitiveness, and corporate reputation. Secondly, for users, paying attention to this can improve users' productivity, reduce training and support costs, improve satisfaction and comfort for users' work, and increase construction investment system efficiency and utilization. Thirdly, service selection study under uncertainty will increase the accuracy of the services provided and give full potential of each service and the effect of cooperating with other services. Finally,

the composite services with uncertainty focus on providing continuity of services to ensure that when QoS changes in a certain range, the composite service is robust and minimizes the impact of service errors or failure.

Over the past few years, cloud computing has been more and more attractive as a new computing paradigm due to high flexibility for provisioning on-demand computing resources that are used as services through the Internet [19, 20]. It is very necessary that cloud users be vigilant while selecting the service providers present in the cloud [21, 22]. The proposed approach in this paper is valid in cloud service selection.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61100166) and Xi'an Science and Technology Program (Grant no. CXY1437(8)).

### References

- [1] L. Z. Zeng, B. Boualem, D. Marlon, J. Kalagnanam, and Q. Z. Sheng, "Quality driven Web services composition," in *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, pp. 411–421, Budapest, Hungary, May 2003.
- [2] Q. Yu, M. Rege, A. Bouguettaya, B. Medjahed, and M. Ouzzani, "A two-phase framework for quality-aware Web service selection," *Service Oriented Computing and Applications*, vol. 4, no. 2, pp. 63–79, 2010.

- [3] C.-W. Zhang, S. Su, and J.-L. Chen, "Genetic algorithm on web services selection supporting QoS," *Chinese Journal of Computer*, vol. 29, no. 7, pp. 1029–1037, 2006 (Chinese).
- [4] T. Wen, G.-J. Sheng, Q. Guo, and Y.-Q. Li, "Web service composition based on modified particle swarm optimization," *Chinese Journal of Computers*, vol. 36, no. 5, pp. 1031–1046, 2013 (Chinese).
- [5] Y.-M. Xia, B. Cheng, J.-L. Chen, X.-W. Meng, and D. Liu, "Optimizing services composition based on improved ant colony algorithm," *Chinese Journal of Computers*, vol. 35, no. 2, pp. 270–281, 2012 (Chinese).
- [6] S.-G. Deng, L.-T. Huang, B. Wu, J.-W. Yin, and G.-X. Li, "QoS optimal automatic composition of semantic Web services," *Chinese Journal of Computers*, vol. 36, no. 5, pp. 1015–1030, 2013 (Chinese).
- [7] G. H. Alférez and V. Pelechano, "Facing uncertainty in Web service compositions," in *Proceedings of the IEEE 20th International Conference on Web Services (ICWS '13)*, pp. 219–226, Santa Clara, Calif, USA, July 2013.
- [8] K. Benouaret, D. Benslimane, and A. Hadjali, "Selecting skyline web services from uncertain QoS," in *Proceedings of the IEEE 9th International Conference on Services Computing (SCC '12)*, pp. 523–530, IEEE, Honolulu, Hawaii, USA, June 2012.
- [9] Q. Wu, Q. Zhu, X. Jian, and F. Ishikawa, "Broker-based SLA-aware composite service provisioning," *Journal of Systems & Software*, vol. 96, no. 4, pp. 194–201, 2014.
- [10] A. Mostafa and M. Zhang, "Multi-objective service composition in uncertain environments," *IEEE Transactions on Services Computing*, 2015.
- [11] F. Wagner, F. Ishikawa, and S. Honiden, "Robust service compositions with functional and location diversity," *IEEE Transactions on Services Computing*, 2014.
- [12] J. K. Zhang, *Optimality and duality for generalized convex uncertain programming [Ph.D. thesis]*, Xidian University, Xi'an, China, 2012, (Chinese).
- [13] A. L. Soyster, "Convexprogramming with set-inclusive constraints and applications to inexact linear programming," *Operations Research*, vol. 21, no. 5, pp. 1154–1157, 1973.
- [14] A. Ben-Tal and A. Nemirovski, "Robust solutions of uncertain linear programs," *Operations Research Letters*, vol. 25, no. 1, pp. 1–13, 1999.
- [15] D. Bertsimas and M. Sim, "Robust discrete optimization and network flows," *Mathematical Programming*, vol. 98, no. 1-3, Ser. B, pp. 49–71, 2003.
- [16] D. Bertsimas and M. Sim, "The price of robustness," *Operations Research*, vol. 52, no. 1, pp. 35–53, 2004.
- [17] M. L. Hu, C. X. Fan, and K. Q. Shi, "Character analysis of standardization methods of decision matrix with intervals," *Computer Science*, vol. 40, no. 10, pp. 203–207, 2013 (Chinese).
- [18] Z. B. Zheng, Y. L. Zhang, and M. R. Lyu, "Distributed QoS evaluation for real-world Web services," in *Proceedings of the IEEE 8th International Conference on Web Services (ICWS '10)*, pp. 83–90, IEEE, Miami, Fla, USA, July 2010.
- [19] Y. Zhong, W. Li, W. Guo, L. Gong, and G. Lodewijks, "A method of modeling and service encapsulation on cloud logistics resources," in *Proceedings of the 19th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD '15)*, pp. 383–388, IEEE, Calabria, Italy, May 2015.
- [20] W. Li, Y. Zhong, X. Wang, and Y. Cao, "Resource virtualization and service selection in cloud logistics," *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1696–1704, 2013.
- [21] P. Bedi, H. Kaur, and B. Gupta, "Trustworthy service provider selection in cloud computing environment," in *Proceedings of the International Conference on Communication Systems and Network Technologies (CSNT '12)*, pp. 714–719, Rajkot, India, May 2012.
- [22] R. Karim, C. Ding, A. Miri, and X. Liu, "End-to-end QoS mapping and aggregation for selecting cloud services," in *Proceedings of the International Conference on Collaboration Technologies and Systems (CTS '14)*, pp. 515–522, IEEE, Minneapolis, Minn, USA, May 2014.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

