Web Service Composition with Uncertain QoS: An IQCP Model

Hengzhou Ye[1,2](✉) and Taoshen Li[3]

[1] Guangxi Key Laboratory of Embedded Technology and Intelligent System,
Guilin University of Technology, No. 12, Jiangan Road, Qixing District,
Guilin 541004, China
2002018@glut.edu.cn
[2] College of Electrical Engineering, Guangxi University,
No. 100, Daxue Street, Xixiangtang District, Nanning 530004, China
[3] School of Information and Engineering, Guangxi University, Nanning, China
tshli@gxu.edu.cn

Abstract. Quality of Service (QoS) is commonly employed to represent non-functional web service (WS) characteristics for the purpose of optimizing WS composition. As a departure from most of the extant research on QoS aggregations, where QoS is typically represented deterministically, we hypothesize the QoS to a WS as a random variable that follows a normal distribution. A serial of formulas are proposed to calculate the expectation and variance of the QoS of a composite service; this yields four QoS criteria suited to workflow described by a directed acrylic graph (DAG). The Web service composition problem with uncertain QoS is then modeled as an integer quadratically constrained program (IQCP). Finally, a series of experimental results obtained in CPLEX and Java illustrate that our model has favorable robustness and can estimate composite service QoS rapidly and accurately.

Keywords: Uncertain QoS · Service composition
Integer quadratically constrained program · Directed acyclic graph

1 Introduction

The coarse-grained, loosely coupled service-oriented architecture (SOA) processes communication among services through simple and well-defined interfaces independent of the underlying implementation platform or network communication module. WS and SOA-based software systems are often combined with various other services to realize SOA. The WS composition problem has received a great deal of research attention, to this effect. With the proliferation of WSs on the Internet, QoS is commonly adopted to describe non-functional WS characteristics. Optimizing the QoS-aware service composition (QSC) is an especially popular research subject in this field. The goal is to select a composite WS that maximizes certain aggregated-quality functions while implementing the desired user functionalities and preserving several QoS constraints.

Y. Sun et al. (Eds.): ChineseCSCW 2018, CCIS 917, pp. 146–162, 2019.
https://doi.org/10.1007/978-981-13-3044-5_11

The workflow-based QSC problem has been extensively studied as well [1–4]. However, most previous researchers consider each WS to have a deterministic QoS. In actuality, the QoS measure of a WS is intrinsically probabilistic due to the complexity and dynamic nature of the network environment [5] and is very challenging to accurately estimate. For example, the response time for a WS is dependent upon the number of requests invoking it. As discussed by Wang et al. [6], the QoS obtained by the service provider's description (or the QoS value calculated through historical information) does not truly reflect the performance of the service. For scientific computing tasks, service oriented applications, and MapReduce applications in cloud environment, research has shown that the CPU, network, and I/O performance may fluctuate significantly in the short term [7, 8]. Armbrust et al. [9] found that the performance of a service can fluctuate by 4–16% due to network and disk I/O interference. The QoS of a WS should be described in an uncertain form in order to ensure an accurate and workable QSC problem model [10].

Previous researchers have represented the QoS of a WS as a single value, multiple values [11], standard statistical distribution [12, 13], and any probability distribution [14]. QoS, when represented as a constant value, does not contain quality variations. It is more reasonable to model QoS as a standard statistical distribution (e.g., normal distribution) than several values with different frequencies. Though not every QoS measure of a WS follows a normal distribution, taking any probability distribution into consideration will increase the difficulty of the problem significantly.

In this study, we assumed that the QoS to a WS follows a normal distribution. In an attempt to design a DAG-based workflow, we established QoS aggregation methods for several QoS criteria and built an IQCP to tackle the QSC problem with uncertain QoS. The main contributions of this work are as follows.

- We use an original and efficient aggregation approach for maximum/min-type and product-type QoSs. Compared to representing QoSs in any probability distribution, the proposed method estimates QoS aggregation quicker and more accurately.
- We built the QSC problem with uncertain QoS into the well-known IQCP model, which is promising for exactly solving the composition problem with uncertain QoS.

The remainder of this paper is structured as follows. An overview of the research on Web service composition with uncertain QoS is conducted in Sect. 2. Section 3 lists the necessary assumptions and theorems. Section 4 describes the workflow and QoS model. Section 5 details QoS aggregation calculation process. Section 6 proposes the WS composition model with uncertain QoS; Sect. 7 describes its performance in detail. Section 8 provides a brief summary and conclusion.

2 Related Work

The global constraint decomposition strategy [15–17] can be adopted to tackle with the QSC problem by considering the uncertainty of QoS. This typically involves dividing the WS composition process into two phases: decomposition of global constraints and local optimization selection. In the former phase, the global constraints are decomposed into a series of constraints imposed on each subtask only. Using these local constraints, the local selection process is carried out via optimization to quickly select best services while ensuring that global constraints are satisfied. When exceptions occur during running time, an appropriate substitution can be quickly identified by simply repeating the local selection process. The strategy is thus adaptable to dynamic environments to a certain extent. Chen et al. [18] proposed the instant recommendation approach to deal with manage uncertain QoS, which works by revealing the most reliable and robust services per the execution log of composite services, therefore, user demands can be fulfilled with higher probability. Hyunyoung et al. [19] also estimated actual QoS performance to a service based on the real transaction history rather than the QoS information published by its provider.

Representing QoSs as multiple values or probability distributions may be a more straightforward way to resolve the uncertain QoS service composition problem. Wang et al. [6] and Shen et al. [20] used the cloud model to evaluate QoS uncertainty; three key parameters(expected value, entropy and hyper entropy) were used to characterize the stability of QoS, then to decrease the number of candidate or composite services, redundant services were pruned by Skyline computing. Skyline computing was also adopted by following work. Fu et al. [21] used an empirical distribution function to describe QoS uncertainty with special focus on stochastic dominance (SD) theory. The method discussed by Fu et al. [22] does not require the assumption that QoS has a specific distribution, and focuses on aggregating the QoS in a cumulative manner. Yu et al. [23] developed the novel p-dominant service skyline concept, which is computed based on a p-R-tree indexing structure and a dual-pruning scheme.

Some researchers have calculated the QoS of a composite service, called QoS aggregation, which is one of the core issues relevant to the QSC problem. Hwang et al. [5] presented a uniform probabilistic model to denote the QoS of atomic or composite WSs with corresponding computation algorithms. The method is precise, but extremely time-consuming. Zheng et al. [14] developed a set of formulas for QoS aggregation according to four typical patterns: sequential, concurrent, selection, and loop. As opposed to the method presented by Hwang et al. [5], its numerical computation algorithms stipulate that the starting point and width of the intervals must be consistent for all QoS distributions – this unfortunately makes QoS monitoring and parameter-setting more difficult. They also ignore QoS aggregation for multiplicative QoS (e.g., reliability) to avoid any combinatorial explosions. Chellammal et al. [15] also denotes QoS denoted as a Probability Mass Function (PMF). By introducing the global constraint decomposition strategy, QoS aggregation is only calculated when the composite service selected via local optimization is unfit for user requests. This reduces the high time overhead on QoS aggregation. By modeling QoS values in normal distribution, Schuller et al. [24] selected the optimized service combination at minimal cost under QoS requirements; they used a simulation approach for QoS estimation. Wang et al. [25] focused on the uncertainty of service execution rather than the uncertainty of QoS. Du et al. [26] and George et al. [27] only used one QoS criterion each: the former used response time, the latter used cost.

3 Underlying Assumptions and Theorems

We held the following assumptions in conducting this study:

(1) The QoS to a WS follows a normal distribution and the QoS of one WS is unrelated to the QoSs of other WSs.

(2) When QoSs to each WS follow normal distributions, the QoS aggregation to a composite service combined by these WSs follows a normal distribution.

(3) For a given workflow F, let $s = (s_1, s_2, \ldots, s_n)$ be an arbitrary composite service to F and the response time of s_i $(i = 1, 2, \ldots, n)$ follow a normal distribution $N\left(\mu_i, \sigma_i^2\right)$. There are two non-negative real number sequences $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$ which can be used to calculate the expectation E(s) and variance D (s) of the response time of s as follows:

$$E(s) = \sum\nolimits_{i=1}^{n} x_i \cdot \mu_i, \quad D(s) = \sum\nolimits_{i=1}^{n} y_i \cdot \sigma_i^2 \tag{1}$$

Assume that X_i is a random variable, and $X_i \sim N\left(\mu_i, \sigma_i^2\right)$ $(i = 1, 2, \ldots, n)$, and X_i is independent of X_j $(i \neq j)$. Let $Y_n = \prod_{i=1}^{n} X_i$. The expectation and variance of Y_n are denoted as E(Y_n) and D(Y_n), respectively.

Theorem 1. $E(Y_n) = \prod_{i=1}^{n} \mu_i$.

Proof: Because $X_1, X_2, \ldots, X_n$ are independent of each other, E(Y_n) can be obtained as follows:

$$E(Y_n) = E\left(\prod_{i=1}^{n} X_i\right) = \prod_{i=1}^{n} E(X_i) = \prod_{i=1}^{n} \mu_i$$

Theorem 2. If $\mu_i = \varepsilon\sigma_i$, then $(Y_n) = \left[\left(1 + \frac{1}{\varepsilon^2}\right)^n - 1\right] \prod_{i=1}^{n} \mu_i^2$

Proof: Apply mathematical induction to n.

(1) When $n = 2$,

$$\begin{aligned} D(Y_2) = D(X_1 X_2) &= \sigma_1^2\sigma_2^2 + \sigma_1^2\mu_2^2 + \sigma_2^2\mu_1^2 \\ &= \left(2\varepsilon^2 + 1\right)\sigma_1^2\sigma_2^2 = \left[\left(1 + \frac{1}{\varepsilon^2}\right)^2 - 1\right]\mu_1^2\mu_2^2 \end{aligned}$$

(2) Let us assume that when n is equal to k, the theorem is true. That is,

$$D(Y_k) = \left[\left(1 + \frac{1}{\varepsilon^2}\right)^k - 1\right] \prod\nolimits_{i=1}^{k} \mu_i^2.$$

When $n = k + 1$,

$$\begin{aligned} D(Y_{k+1}) &= D(Y_k)D(X_{k+1}) + D(Y_k)[E(X_{k+1})]^2 + D(X_{k+1})[E(Y_k)]^2 \\ &= \left\{\left[\left(1+\frac{1}{\varepsilon^2}\right)^k - 1\right]\prod_{i=1}^{k}\mu_i^2\right\}(\sigma_{k+1}^2 + \mu_{k+1}^2) + \sigma_{k+1}^2\prod_{i=1}^{k}\mu_i^2 \\ &= \left[\left(1+\frac{1}{\varepsilon^2}\right)^{k+1} - 1\right]\prod_{i=1}^{k+1}\mu_i^2 \end{aligned}$$

These two steps yield the conclusion.

4 Workflow and QoS Model

A workflow represents how to constitute the capabilities of different WSs in four basic patterns (sequence, concurrency, selection, and loop). The labeled graph [28], numbered graph [20], and DAG [29] are common ways to represent a workflow. In the resource allocation field, DAG is used to represent the workflow [30, 31]. DAG cannot directly denote the selection and loop patterns. However, the loop pattern can be regarded as a special sequential one. A workflow with selection patterns can be broken up into several workflows without any selection pattern. Therefore, a workflow with selection and loop patterns can be transformed into several workflows that can be represented by DAG. Consider the workflow shown in Fig. 1, which can be split into the two workflows shown in Fig. 2a and b, respectively. Here, we only consider workflows that can be represented with DAG.

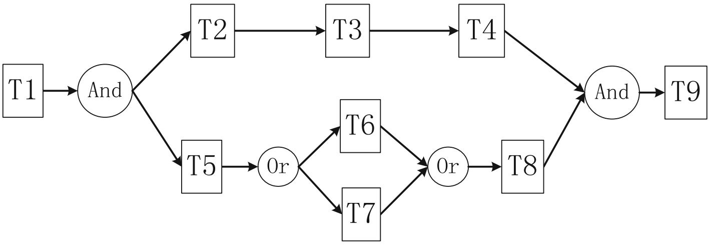
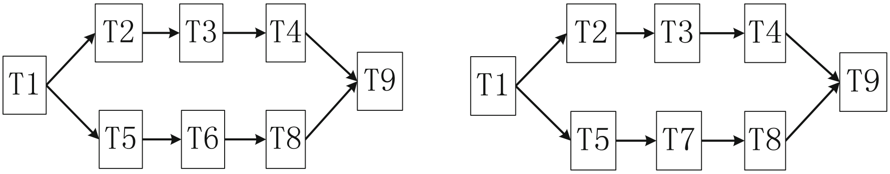


Fig. 1. Workflow with selection pattern



(a) Workflow when T6 is chosen

(b) Workflow when T7 is chosen

Fig. 2. Equivalent workflow to Fig. 1 represented by DAG

The QoS is used to measure the performance of candidate services. The most commonly used QoS criteria include cost, response time, reliability, availability, reputation, and throughput. According to the aggregation method, these criteria can be divided into four classes: sum-type (e.g., cost), min/max-type (e.g., response time), product-type (e.g., reliability), and average-type (e.g., reputation). Similar to [14], the aggregation rules for sequence and concurrency patterns and different types of QoS criteria are summarized in Table 1.

5 Expectation and Variance of QoS for Composite Services

Assume that there are n tasks $T = \{T_1, T_2, \ldots, T_n\}$ in a workflow F. Each task *Ti*, $i \in [1, n]$ has m number of candidate WSs $s_i = \{s_{i1}, s_{i2}, \ldots, s_{im}\}$. A set of 0–1 variables $\boldsymbol{x} = \{x_{ij}\}(1 \le i \le n,\ 1 \le j \le m)$ represent a combination $cs(x)$ of F. When the task t_i chooses the service s_{ij}, $p_{ij} = 1$, otherwise $p_{ij} = 0$. Let the cost, response time, reliability, and reputation of s_{ij} follow normal distributions $\mathrm{N}\left(\mu p_{ij}, \sigma p_{ij}^2\right)$, $\mathrm{N}\left(\mu t_{ij}, \sigma t_{ij}^2\right)$, $\mathrm{N}\left(\mu r_{ij}, \sigma r_{ij}^2\right)$ and $\mathrm{N}\left(\mu c_{ij}, \sigma c_{ij}^2\right)$, respectively. According to our assumptions, the cost, response time, reliability, and reputation of $cs(x)$ will also follow normal distributions. Their corresponding expectation and variance are discussed below.

Table 1. Aggregation rules for different patterns and types of QoS

QoS type	Pattern	Aggregation rules
Sum-type	Sequence/concurrency	Addition of their QoS values
Min/max-type	Sequence concurrency	Addition of their QoS values maximum of their QoS values
Product-type	Sequence/concurrency	Multiplication of their QoS values
Average-type	Sequence/concurrency	Average of their QoS values

5.1 Expectation and Variance of Cost

According to Table 1, the cost of a composite service can be summed by the cost of all its components. The linear combination of a set of independent normal random variables still obeys a normal distribution, so the cost of $cs(x)$ is distributed from a normal distribution. Thus, the expectation $\mathrm{E}_p(cs(x))$ and variance $\mathrm{D}_p(cs(x))$ of $cs(x)$ can be obtained by Formulas (2) and (3), respectively:

$$\mathrm{E}_p(cs(x)) = \sum\nolimits_{i=1}^{n} \sum\nolimits_{j}^{m} p_{ij} \cdot \mu p_{ij} \tag{2}$$

$$\mathrm{D}_p(cs(x)) = \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \sigma p_{ij}^2 \tag{3}$$

5.2 Expectation and Variance of Response Time

Let j represent an arbitrary composite service of F which consists of a series of services $(s_{1j_1}, s_{1j_1}, \ldots, s_{nj_n})$. Under our assumptions, the existence of two non-negative real number sequences $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$ yields the following two formulas:

$$\sum\nolimits_{i=1}^{n} x_i \cdot \mu t_{1j_1} = \mu t_j, \quad \sum\nolimits_{i=1}^{n} y_i \cdot \sigma t_{1j_1}^2 = \sigma t_j^2 \tag{4}$$

where μt_j, σt_j denote the expectation and mean square deviation of j, respectively. Their values can be calculated as follows by sampling:

$$\mu t_j = \sum\nolimits_{i=1}^{Times} t_{ij_i}(k), \quad \sigma t_j^2 = \sum\nolimits_{i=1}^{Times} \left[t_{ij_i}(k) - \mu t_j\right]^2 \tag{5}$$

where $t_{ij_i}(k)$ denotes the k-th response time of s_{ij_i} and Times is the number of sampling iterations.

Select q number of different composite services $(j_1, j_2, \ldots, j_q)$ by random for F and let:

$$x = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}, \quad u = (\mu t_{j1}, \mu t_{j2}, \ldots, \mu t_{jn})^{\mathrm{T}} \tag{6}$$

$$U = \begin{pmatrix} \mu t_{1j_{11}} & \mu t_{2j_{12}} & \cdots & \mu t_{nj_{1n}} \\ \mu t_{1j_{21}} & \mu t_{2j_{22}} & \cdots & \mu t_{nj_{2n}} \\ \cdots & \cdots & \cdots & \cdots \\ \mu t_{1j_{q1}} & \mu t_{2j_{q2}} & \cdots & \mu t_{nj_{qn}} \end{pmatrix} \tag{7}$$

yielding the following expression:

$$Ux = u \tag{8}$$

When $q > n$, Formula (8) is a non-negative overdetermined linear equation system. Its solution, that is, the value of x, can be calculated by a known method.

Similarly, let:

$$y = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}, \quad o = (\sigma t_{j1}, \sigma t_{j2}, \ldots, \sigma t_{jn})^{\mathrm{T}} \tag{9}$$

$$O = \begin{pmatrix} {\sigma t_{1j_{1_1}}}^2 & {\sigma t_{2j_{1_2}}}^2 & \cdots & {\sigma t_{nj_{1_n}}}^2 \\ {\sigma t_{1j_{2_1}}}^2 & {\sigma t_{2j_{2_2}}}^2 & \cdots & {\sigma t_{nj_{2_n}}}^2 \\ \cdots & \cdots \quad \cdots & & \cdots \\ {\sigma t_{1j_{q_1}}}^2 & {\mu t_{2j_{q_2}}}^2 & \cdots & {\sigma t_{nj_{q_n}}}^2 \end{pmatrix} \tag{10}$$

This allows us to obtain following equation:

$$Oy = o \tag{11}$$

y can be obtained by solving Eq. (11).

Calculating the value of x or y are time consuming. However, the calculation process can be completed offline because it depends only on the workflow and candidate services, and is independent of user requirements. Hence, this process does not affect the time overhead of combining services.

After determining values of x and y, the expectation and variance of the response time of $cs(x)$, denoted as $\mathrm{E}_t(cs(x))$ and $\mathrm{D}_t(cs(x))$, respectively, can be calculated as follows:

$$\mathrm{E}_t(cs(x)) = \sum\nolimits_{i=1}^{n} \left(x_i \sum\nolimits_{j=1}^{m} p_{ij} \cdot \mu t_{ij} \right), \quad \mathrm{D}_t(cs(x)) = \sum\nolimits_{i=1}^{n} \left(y_i \sum\nolimits_{j=1}^{m} p_{ij} \cdot \sigma t_{ij}^2 \right) \tag{12}$$

5.3 Expectation and Variance of Reliability

According to Table 1, the reliability of a composite service can be achieved by multiplying the reliability of all its components. Based on Theorem 1, the expectation of reliability of $cs(x)$, denoted as $\mathrm{E}_r(cs(x))$, can be calculated as follows:

$$\mathrm{E}_r(cs(x)) = \prod\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \mu r_{ij} \tag{13}$$

According Theorem 2, the variance of reliability of $cs(x)$, denoted as $\mathrm{D}_r(cs(x))$, can be calculated approximately by Formula (14):

$$\mathrm{D}_r(cs(x)) = \left[\left(1 + \frac{1}{\varepsilon^2} \right)^n - 1 \right] \prod\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \mu r_{ij}^2 \tag{14}$$

where the parameter ε can be obtained as follows:

$$\varepsilon = \frac{1}{n \cdot m} \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} \frac{\mu r_{ij}}{\sigma r_{ij}} \tag{15}$$

5.4 Expectation and Variance of Reputation

According to Table 1, the reputation of a composite service can be calculated by averaging the reputation of all its components. The expectation and variance of the reputation of $cs(x)$, denoted as $\mathrm{E}_c(cs(x))$ and $\mathrm{D}_c(cs(x))$, respectively, can be calculated as follows:

$$\mathrm{E}_c(cs(x)) = \frac{1}{n}\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{m} p_{ij}\cdot\mu c_{ij}, \quad \mathrm{D}_c(cs(x)) = \frac{1}{n^2}\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{m} p_{ij}\cdot\sigma c_{ij}^2 \tag{16}$$

6 Web Service Composition Model with Uncertain QoS

Without loss of generality, our aim is to minimize the cost while satisfying QoS constraints in regards to response time, reliability, and reputation. Our model is described in detail below.

$$\text{Object:} \quad \min\left(\sum\nolimits_{i=1}^{n}\sum\nolimits_{j}^{m} p_{ij}\cdot\mu p_{ij} + \beta\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{m} p_{ij}\cdot\sigma p_{ij}^2\right) \tag{17}$$

$$\text{s.t.}: \quad P(q_t \le C_t) \ge p_t, \quad P(q_r \ge C_r) \ge p_r, \quad P(q_c \ge C_c) \ge p_c \tag{18}$$

$$p_{ij} \in \{0,1\}, 1 \le i \le n, 1 \le j \le m \tag{19}$$

$$\sum\nolimits_{j=1}^{m} p_{ij} = 1, 1 \le i \le n \tag{20}$$

where β is a tunable parameter, $P(X \le x)$ is the probability that the value of X falls into the interval $(-\infty, x]$, C_t, C_r, and C_c respectively represent the constraints of response time, reliability, and reputation, and p_t, p_r, and p_c are given constants.

Considering that the QoS of composition services are subject to normal distributions, Inequation (18) can be converted into Inequations (21)–(23) in accordance with the 3$\boldsymbol{\sigma}$ principle:

$$\mu_t + 3\sigma_t \le C_t \tag{21}$$

$$\mu_r - 3\sigma_r \ge C_r \tag{22}$$

$$\mu_c - 3\sigma_c \ge C_c \tag{23}$$

where μ_t, σ_t, μ_r, σ_r, μ_c, and σ_c respectively represent the expectation and mean variance of the response time, reliability, and reputation of a composite service.

Inequation (21) is equivalent to the following two inequations:

$$0 \le C_t - \mu_t \tag{24}$$

$$9\sigma_t^2 \le (C_t - \mu_t)^2 \tag{25}$$

Substituting Formula (12) into in Inequation (25), and introducing the tunable parameters β_1 and β_2 (considering that there exists some error in the expectation and variance of the response time), yields the following:

$$9\beta_2 \sum\nolimits_{i=1}^{n} \left(y_i \sum\nolimits_{j=1}^{m} p_{ij} \cdot \sigma t_{ij}^2 \right) \le \left(C_t - \beta_1 \sum\nolimits_{i=1}^{n} \left(x_i \sum\nolimits_{j=1}^{m} p_{ij} \cdot \mu t_{ij} \right) \right)^2 \tag{26}$$

Introduce a variable $\gamma = \sqrt{\left(1 + \frac{1}{\varepsilon^2}\right)^n - 1}$, Substituting Formulas (13) and (14) into inequation (22), and introducing the tunable parameters β_3 (considering that there exists some error in the variance of the reliability), yields the following:

$$\prod\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \mu r_{ij} - 3\beta_3 \gamma \sqrt{\prod\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \mu r_{ij}^2} \ge C_r \tag{27}$$

Note that $\sqrt{p_{ij}} = p_{ij}$. After some simplifications, Inequation (27) is equivalent to the following inequality:

$$(1 - 3\beta_3 \gamma) \cdot \prod\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \mu r_{ij} \ge C_r \tag{28}$$

Taking the logarithm of both sides of Inequality (28) yields the following:

$$\sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \log\left(\mu r_{ij}\right) \ge \log(C_r/(1 - 3\beta_3 \gamma)) \tag{29}$$

Inequality (23) is equivalent to the following condition:

$$\mu_c - C_c \ge 0, \quad (\mu_c - C_c)^2 \ge 9\sigma_c^2 \tag{30}$$

Substituting Formula (16) into Inequality (30) yields:

$$\left(\frac{1}{n} \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \mu c_{ij} - C_c \right)^2 \ge \frac{3}{n^2} \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \sigma c_{ij}^2 \tag{31}$$

In summary, the problem with WS composition for uncertain QoS can be represented as an IQCP model.

$$\text{Object:} \quad \min\left(\sum\nolimits_{i=1}^{n} \sum\nolimits_{j}^{m} p_{ij} \cdot \mu p_{ij} + \beta \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \sigma p_{ij}^2 \right) \tag{32}$$

$$\text{s.t.}: \quad 0 \le \mathrm{C_t} - \upbeta_1 \sum\nolimits_{i=1}^{n} \left(x_i \sum\nolimits_{j=1}^{m} p_{ij} \cdot \mu t_{ij} \right) \tag{33}$$

$$9\upbeta_2 \sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} \left(\mathrm{y_i} \sum\nolimits_{\mathrm{j}=1}^{\mathrm{m}} \mathrm{p_{ij}} \cdot \upsigma \mathrm{t_{ij}^2} \right) \le \left(\mathrm{C_t} - \upbeta_1 \sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} \left(\mathrm{x_i} \sum\nolimits_{\mathrm{j}=1}^{\mathrm{m}} \mathrm{p_{ij}} \cdot \upmu \mathrm{t_{ij}} \right) \right)^2 \tag{34}$$

$$\sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \log\left(\mu r_{ij}\right) \ge \log(C_r/(1 - 3\beta_3 \gamma)) \tag{35}$$

$$\sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} p_{ij} \cdot \mu c_{ij} - n \cdot C_c \ge 0 \tag{36}$$

$$\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \mathrm{p_{ij}}\cdot \mu \mathrm{c_{ij}} - \mathrm{n}\cdot \mathrm{C_c}\right)^2 \geq 3\cdot \sum_{i=1}^{n}\sum_{j=1}^{m} \mathrm{p_{ij}}\cdot \sigma \mathrm{c_{ij}^2} \tag{37}$$

$$p_{ij} \in \{0,1\}, 1 \leq i \leq n, 1 \leq j \leq m \tag{38}$$

$$\sum_{j=1}^{m} p_{ij} = 1,\ 1 \leq i \leq n \tag{39}$$

7 Experiments

7.1 Robustness Metrics

Several previous researchers have explored temporal robustness metrics for resource scheduling or service composition problems. There is no consensus on which metric should be adopted, but instead it is up to the scholar's discretion per the problem at hand. Tolerance time [30], makespan mean [31], slack time [32], and robustness probability [33] are commonly used metrics. In this study, we established the following two metrics according to these metrics.

The first is robust probability R_p, which represents the probability that the selected composite service satisfies the stated constraints. Let *TotalTimes* represent the total number of tests and *FailedTimes* represent the total number of defaults. R_p is calculated as follows:

$$R_p = (\mathrm{TotalTimes} - \mathrm{FailedTimes})/\mathrm{TotalTimes} \tag{40}$$

The other is relaxation metrics R_s, which represents the gap between user constraints and the aggregated QoS of the selected composite service:

$$R_s = (C_t - t)/C_t + (r - C_r)/C_r + (c - C_c)/C_c \tag{41}$$

where C_t, C_r, and C_c respectively denote the restrictive conditions of response time, reliability, and reputation, while t, r, and c respectively denote the response time, reliability, and reputation of the selected composite service. The values of t, r, and c are random, so the R_s value is the minimum value of multiple measurements.

7.2 Simulation Environment and Parameter Settings

We conducted experiments on a PC which has a 2.4 GHz CPU and 4 GB of memory installed with win7 and JRE6. We used CPLEX to solve the IQCP model and the function lsqnonneg in Matlab to solve the non-negative overdetermined linear equation system. The expectations of response time, reliability, and reputation to a candidate service were taken from the QWS database [34]. The expectation of cost was randomly evaluated on the interval [100, 200] due to the lack of information about cost in this database. As pointed out by Armbrust et al. [9], the fluctuation range of response time can reach 4–16%. For the response time, we let the mean variance be times the expectation where is a random value on [0.1, 0.2]. We took a similar approach to the mean variance of cost and reputation. The reliability criterion belongs to product-type. If the magnitude of fluctuation is relatively large, the reliability to a composite service including a lot of component services may tend towards zero. Thus, the reliability criterion is a random value on [0.001, 0.015]. And the maximum reliability and reputation criteria were set to 1.

If a candidate service is selected for each task of the workflow, its QoS is the average expectation of the QoS for all its candidate services. The response time, reliability, and reputation of this composite service are denoted as BC_t, BC_r, and BC_c, respectively, then the values of C_t, C_r, and C_c are set to 1.2 * BC_t, 0.8 * BC_r, and 0.8 * BC_c, respectively. We set the number of samples to 10000, $\beta = 0$, and $\beta_1 = \beta_2 = \beta_3 = 1$.

The DAGs in our experiments were randomly generated. The number of nodes starts at 10 and increases to 100 in intervals of 10. In DAG, there is an initial node and a termination node. Each node has 1–4 direct child nodes except the termination node at a ratio of 6:3:2:1. The number of candidate services per task also starts at 10 and increases to 100 by intervals of 10.

7.3 Robustness Analysis

When number of tasks was assigned 20 and number of WSs varied between 10 and 100, values of R_p and R_s were as shown in Table 2. Table 3 shows these values when number of WSs was assigned 20 and number of tasks varied between 10 and 100. The R_p values in both tables are approximately 99.9% for different number of tasks and WSs, i.e., more than 99.74% as determined by the 3**σ** principle. The value of R_s is around 0.7, indicating that there were still some gaps between user constraints and the aggregated QoS of the selected composite service in our experiment. The values of R_p and R_s were less affected by the scale of the problem, indicating that our model has good stability.

Table 2. R_p and R_s over WSs with 20 tasks

Number of WSs	R_p	R_s
10	0.9995	0.7008
20	0.9995	0.6452
30	0.9992	0.6885
40	0.9997	0.6893
50	0.9996	0.6932
60	0.9993	0.6549
70	0.9995	0.6573
80	0.9994	0.6243
90	0.9989	0.5602
100	0.9990	0.7446

Table 3. R_p and R_s over tasks with 20 WSs

Number of tasks	R_p	R_s
10	0.9997	0.7198
20	0.9981	0.6110
30	0.9989	0.6946
40	0.9990	0.6536
50	0.9993	0.5923
60	0.9995	0.6926
70	0.9978	0.6191
80	0.9989	0.6471
90	0.9996	0.7206
100	0.9999	0.7221

The results shown in Figs. 3 and 4 indicate that the time overhead increases rapidly with the number of tasks and the number of services when using CPLEX. More efficient algorithms are yet necessary.

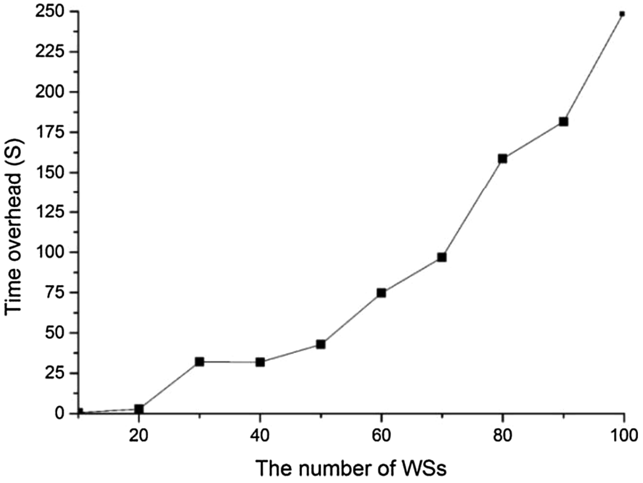


Fig. 3. Time overhead over a range of WSs with 20 tasks

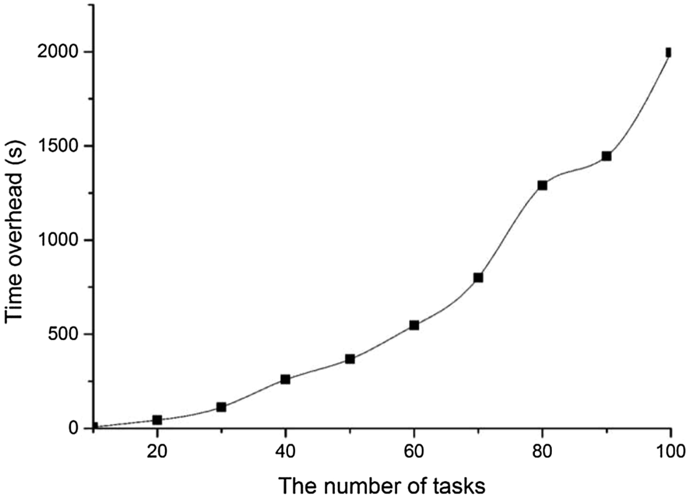
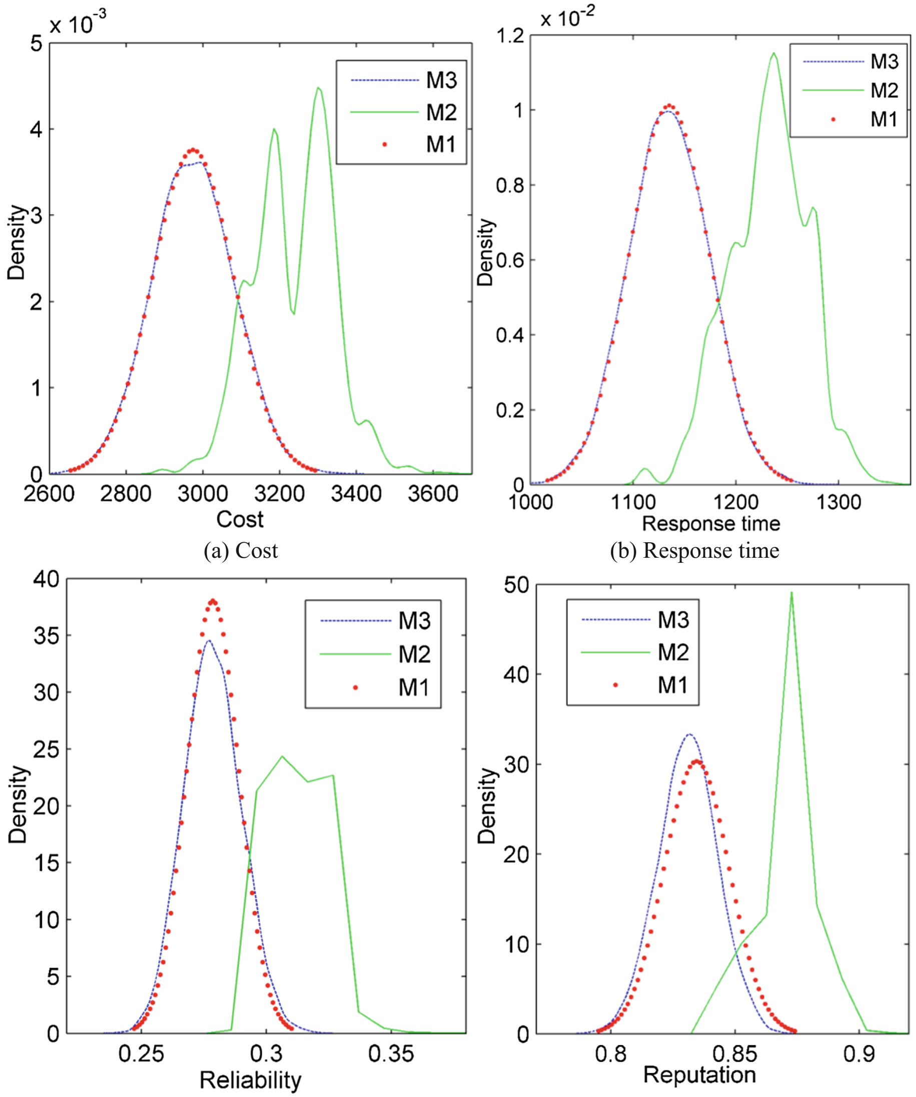


Fig. 4. Time overhead over a range of tasks with 20 WSs

7.4 QoS Estimation of Composite Services

Accurate and rapid estimation of QoS is the key to resolving the large-scale WS composition problem with uncertain QoS. We evaluated the QoS distribution and time overhead of our approach (labeled as M1) compared to the method adopted by Hwang et al. [5] (labeled as M2) and the simulation method adopted by Zheng et al. [14] (labeled as M3). The number of tasks and WSs are assigned 20 and 100, respectively.



(a) Cost

(b) Response time

(c) Reliability

(d) Reputation

Fig. 5. QoS distribution to composite services for three methods

For M2, we adopted the algorithm and parameters recommended by Hwang et al. [5]; that is, the aggregate random variable discovery problem (ARVD) used the greedy strategy, the sample space of a single random variable was set to 20, and the aggregate size of the sample space was set to 30. For M3, number of samples was 10000.

We estimated the QoS distribution for any composite service for a given workflow with 20 tasks using the above three methods; the results are shown in Fig. 5. Generally, when the number of samples was large enough, the results obtained by M3 were very close to the actual. The distributions of cost (Fig. 5a), response time (Fig. 5b), reliability (Fig. 5c), and reputation (Fig. 5d) obtained by our method were approximately the same as M3. The results obtained by M2 deviated substantially.

As shown in Fig. 6, the time complexity of M1 was far less than M2 or M3 for the number of tasks. In effect, our method is better suited to solving large-scale service composition problems with uncertain QoSs.

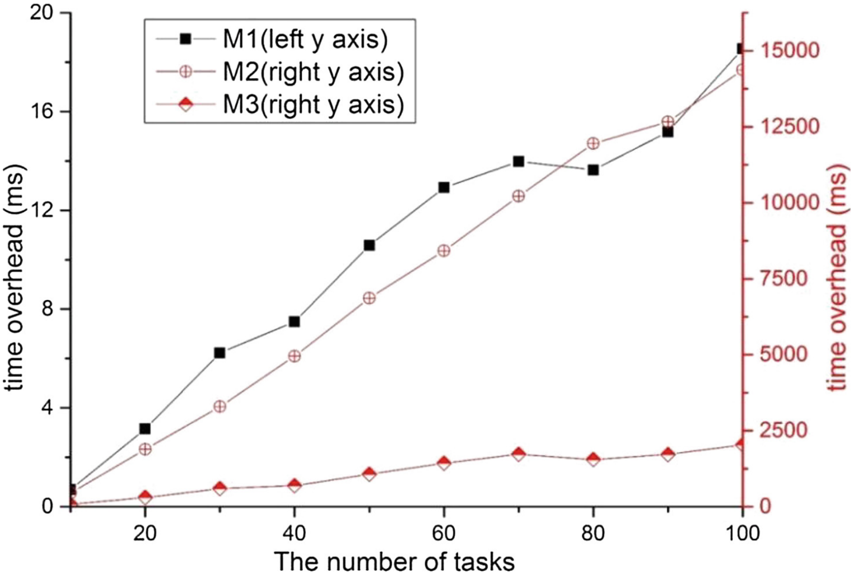


Fig. 6. Time overhead to calculate QoS for three methods

8 Conclusions

As distributed and integrated applications, WSs are invoked over a network (usually the Internet). The corresponding QoS is affected by many factors including the network environment, hardware facilities, user behavior, and others, making it very challenging to accurately estimate. The model used to describe the WS composition problem with uncertain QoS must be sufficiently robust – in other words, the selected composite services should have a high probability of meeting user requirements even if the QoSs of WSs are volatile. In this study, we represented the WS composition problem with uncertain QoS as an IQCP model based on some assumptions and approximations. We validated the proposed model by a series of simulations.

In the future, we plan to further optimize the IQCP model and its parameters. We also plan to find more effective algorithms to solve the model and to apply to other types of QoS probability distributions.

Acknowledgment. This work is supported by Guangxi Universities key Laboratory Director Fund of Embedded Technology and Intelligent Information Processing (Guilin University of Technology) under Grand No. 2018A-05.

References

1. Ramírez, A., Parejo, J.A., Romero, J.R., et al.: Evolutionary composition of QoS-aware web services: a many-objective perspective. Expert Syst. Appl. **72**, 357–370 (2017)
2. Zhou, J., Yao, X.: A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition. Int. J. Adv. Manuf. Technol. **88**(9–12), 3371–3387 (2017)
3. Zou, G.B., Lu, Q., Chen, Y.X., et al.: QoS-aware dynamic composition of web services using numerical temporal planning. IEEE Trans. Serv. Comput. **7**, 18–31 (2014)
4. Karimi, M.B., Isazadeh, A., Rahmani, A.M.: QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm. J. Supercomput. **73**(4), 1387–1415 (2017)
5. Hwang, S.Y., Wang, H.J., Tang, J., et al.: A probabilistic approach to modeling and estimating the QoS of web-services-based workflows. Inf. Sci. **177**, 5484–5503 (2007)
6. Wang, S.G., Sun, Q.B., Zhang, G.W., et al.: Uncertain QoS-aware skyline service selection based on cloud model. J. Softw. **23**(6), 1397–1412 (2012)
7. Alexandru, I., Simon, O., Nezih, Y., et al.: Performance analysis of cloud computing services for many-tasks scientific computing. IEEE Trans. Parallel Distrib. Syst. **22**(6), 931–945 (2011)
8. Jiang, D.J., Guillaume, P., Chi, C.H.: Ec2 performance analysis for resource provisioning of service oriented applications. In: Proceedings of the 2009 International Conference on Service-Oriented Computing, pp. 197–207 (2009)
9. Armbrust, M., Fox, A., Griffith, R., et al.: A view of cloud computing. Commun. ACM **53** (4), 50–58 (2010)
10. Li, Z., Yang, F.C., Su, S.: Fuzzy multi-attribute decision making-based algorithm for semantic web service composition. J. Softw. **20**(3), 583–596 (2009)
11. Hwang, S.Y., Hsu, C.C., Lee, C.H.: Service selection for web services with probabilistic QoS. IEEE Trans. Serv. Comput. **8**(3), 467–480 (2015)
12. Zhu, X.L., Wang, B.: Web service selection algorithm based on uncertain quality of service. Comput. Integr. Manuf. Syst. **17**(11), 2532–2539 (2011)
13. Kattepur, A., Georgantas, N., Issarny, V.: QoS composition and analysis in reconfigurable web services choreographies. IEEE Int. Confer. Web Serv. **125**(3), 235–242 (2013)
14. Zheng, H.Y., Yang, J., Zhao, W.L.: Probabilistic QoS aggregations for service composition. ACM Trans. Web **10**(2), 1–34 (2016)
15. Chellammal, S., Gopinath, G., Manikandan, S.R.: An approach for selecting best available services through a new method of decomposing QoS constraints. SOCA **9**(2), 107–138 (2015)

16. Liu, Z.Z., Xue, X., Shen, J.Q., et al.: Web service dynamic composition based on decomposition of global QoS constraints. Int. J. Adv. Manuf. Technol. **69**(9), 2247–2260 (2013)
17. Ye, H.Z., Li, T.S., Jing, C.: Decomposition of global constraints for QoS-aware web service composition. Int. J. Innov. Comput. Inf. Control **12**(6), 2053–2066 (2016)
18. Chen, L., Wu, J., Jian, H.Y., et al.: Instant recommendation for web services composition. IEEE Trans. Serv. Comput. **7**(4), 586–598 (2014)
19. Hyunyoung, K., Reeseo, C., Wonhong, N.: Transaction history-based web service composition for uncertain QoS. Int. J. Web Grid Serv. **12**, 42–62 (2016)
20. Shen, L.M., Chen, Z., Li, F.: Service selection approach considering the uncertainty of QoS data. Comput. Integr. Manuf. Syst. **19**(10), 2652–2663 (2013)
21. Fu, X.D., Yue, K., Liu, L., et al.: Discovering admissible web services with uncertain QoS. Front. Comput. Sci. **9**(2), 265–279 (2015)
22. Fu, X.D., Yue, K., Liu, L., et al.: Admissible composition plans of web service with uncertain QoS. Comput. Integr. Manuf. Syst. **22**, 122–132 (2016)
23. Yu, Q., Bouguettaya, A.: Computing service skyline from uncertain QoWS. IEEE Trans. Serv. Comput. **3**, 16–29 (2010)
24. Schuller, D., Lampe, U., Eckert, J., et al.: Cost-driven optimization of complex service-based workflows for stochastic QoS parameters. In: IEEE International Conference on Web Services, pp. 66–73 (2012)
25. Wang, P.W., Ding, Z.J., Jiang, C.J., et al.: Automatic web service composition based on uncertainty execution effects. IEEE Trans. Serv. Comput. **9**(4), 551–565 (2016)
26. Du, Y.H., Tan, W., Zhou, M.C.: Timed compatibility analysis of web service composition: a modular approach based on Petri nets. IEEE Trans. Autom. Sci. Eng. **11**(2), 594–606 (2014)
27. George, M., Ioannis, R.: Cost-sensitive probabilistic contingent planning for web service composition. Int. J. Artif. Intell. Tools **25**, 1–20 (2016)
28. Farhad, M., Naser, N., Kamran, Z., et al.: QoS decomposition for service composition using genetic algorithm. Appl. Soft Comput. **6**(5), 3409–3421 (2013)
29. Gabrel, V., Manouvrier, M., Murat, C.: Web services composition: complexity and models. Discrete Appl. Math. **196**(2), 100–114 (2015)
30. Ding, Y.S., Yao, G.S., Hao, K.R.: Fault-tolerant elastic scheduling algorithm for workflow in cloud systems. Inf. Sci. **393**, 47–65 (2017)
31. Chirkin, A.M., Belloum, A.S.Z., Kovalchuk, S.V., et al.: Execution time estimation for workflow scheduling. Future Gener. Comput. Syst. **75**, 376–387 (2017)
32. Deepak, P., Saurabh, K.G., Rajkumar, B., et al.: Robust scheduling of scientific workflows with deadline and budget constraints in clouds. In: Proceedings of the 2014 IEEE 28th International Conference on Advanced Information Networking and Applications, pp. 858–865 (2014)
33. Mark, A.O., Sudeep, P., Anthony, M., et al.: Makespan and energy robust stochastic static resource allocation of a bag-of-tasks to a heterogeneous computing system. IEEE Trans. Parallel Distrib. Syst. **26**(10), 2791–2805 (2015)
34. Eyhab, A.M., Qusay, M.H.: QoS-based discovery and ranking of web services. In: Proceedings of the International Conference on Computer Communications and Networks, pp. 529–534 (2007)