QoS-Based Web Service Composition by GA Using Consumer Decision-Making Function

Gang Wang¹, Li Zhang¹, Wei Jiang¹ and Han Zhaogang²
Department of Computer Science and Engineering, Beihang University¹
Commercial Aircraft Corporation of China, Ltd.²
Beijing, P.R.China
e-mail: f_lag@cse.buaa.edu.cn

Abstract—There are various approaches for QoS-based web service composition using GA (Genetic Algorithm). Most of them use the sum of the values of QoS properties times their weights as the fitness function of GA, to evaluate the QoS of service compositions. However, by using this function, the service composition calculated by GA has shortcomings. For example, response time of the produced service composition is low, good for QoS, and availability is also low, bad for QoS sometimes. It influences users’ QoS experiences. The problem is the function just considers overall QoS while ignoring the equilibrium of QoS properties’ contributions to overall QoS. To deal with this problem, referring to the theory of customer behavior, we use the Cobb-Douglas function indicating users’ decision-making behavior as the fitness function in GA. In this way, the service composition not only has high overall QoS, but each QoS property’s contribution to overall QoS is more balanced.

Keywords— web service composition, QoS, GA, Cobb-Douglas function

I. INTRODUCTION

SOA (Service-Oriented Architecture) becomes a popular development pattern for information system, by coupling various web services loosely and fast to adapt their business processes to meet new requirements. Web services as a basic technology for SOA have been more and more used. With various web services provided, selection of web services for composition becomes an important research topic. With constraints of users’ QoS, selection from the candidate set of web service of each subtask is constraint satisfaction problem. There are several QoS-based approaches of web service composition using GA (Genetic Algorithm) [1-6], which can solve global optimization problems rapidly. GA evaluates the QoS of service compositions by defining the fitness function. QoS requirements have two aspects: values and preferences. So the fitness function is usually defined as the sum of the normalized values of QoS properties times their weights as the fitness function of GA. We call it the product-addition function. However, by using this function, the service composition calculated by GA has shortcomings. For example, response time of the service composition produced by GA is low, is good for QoS. But availability produced is also low, bad for QoS sometimes. It means the service composition responses very quickly, but the possibility of unavailability of the service composition is also very high. It severely influences users’ QoS experiences. The problem is the function just considers overall QoS while ignoring the equilibrium of QoS properties’ contributions to overall QoS. It means current researches are concerned about the maximum of the fitness function while ignoring the equilibrium of each QoS property’s contribution to the fitness function, which cause GA may find the service composition of which one or partial QoS properties are very good. To deal with this problem, referring to the theory of customer behavior, we use the Cobb-Douglas function indicating users’ decision-making behavior as the fitness function in GA. In this way, the service composition not only has high overall QoS, but each QoS property’s contribution to overall QoS is more balanced.

The remainder of this paper is organized as follows. Section 2 details the QoS-based web service composition by GA using consumer decision-making function. Section 3 argues about the advantages of GA using the Cobb-Douglas function, in contrast to GA using the product-addition function. Finally, Section 4 concludes.

II. SEARCH FOR OPTIMIZATION SOLUTIONS OF WEB SERVICE COMPOSITION BY GA USING COBB-Douglas FUNCTION

With constraints of users’ QoS and cost, selection from the candidate set of web service of each subtask is constraint satisfaction problem. GA is used to search for optimization solutions of web service composition, for GA can solve global optimization problems rapidly. We explain our approaches according to the steps of GA specialization for QoS-based web service composition.

The optimization objective is to find optimal solutions of web service composition which has the best trade-off, known as a Pareto set. It indicates best balance between high quality and low cost within the constraints users set.

The implementation of genetic algorithms specific for optimal composition solutions includes the following steps: (1) genome coding, (2) population initialization, (3) fitness evaluation, (4) genetic manipulation

a) Genome coding.
First we need to design a genome in GA to represent the
problem.

Each gene $g_i$ represents a subtask $t_i$. And the $g_i$ takes the integral value from 1 to $n_1$ the number of candidate web services for the subtask $t_i$. We label candidate web services for a subtask from 1 to $n_1$. If value of the gene is $j$, $0<j\leq n_1$, it means selection of web service $j$ to complete the subtask $t_i$. Fig. 1 shows genome coding.

![Figure 1. Genome coding](image)

b) **Fitness evaluation.**

To measure individual the degree near optimal solution, we need to define a fitness function. Referring to the theory of customer behavior, we choose the Cobb-Douglas function [7] as fitness function, which evaluates the composite solutions is the best solutions.

Formula 1 is the function prototype.

$$u(x, y) = x^ay^b, a > 0, b > 0$$  \(1\)

Cobb-Douglas function is one of the most widely used utility function, with its special feature which reflects users’ preference have the following characteristics: First, decision maker think that the more value of the property is, the better the solution is, some properties which is negative correlative with the utility of $u$ such as cost we can use methods to make it positive correlative, secondly, for a specific value $u$, it is better for all properties contribution to $u$ as equal as possible than one or several properties contribution to $u$ far more than others. Because these features reflect characteristics of thinking when making decisions in our daily lives, this paper uses the Cobb-Douglas utility function as decision-making function.

Before using Cobb-Douglas function, data should be preprocessed. Different properties of QoS and cost of web services have different dimension. In our present system, we consider three QoS (response time, availability, reliability) and cost. To remove the effect of different dimensions, data should be normalized to make different properties into the same extent of 0 to 1.

For the positive correlative properties, this paper uses the following normalization function

$$np_{ij} = (p_i - p_{i_{min}})/(p_{i_{max}} - p_{i_{min}})$$  \(1\)

For the negative correlative properties, this paper uses the following normalization function

$$np_{ij} = (p_{i_{max}} - p_i)/(p_{i_{max}} - p_{i_{min}})$$  \(2\)

where $p$ represent a property of QoS and cost of web services, $p_{ij}$ is the value of a property of QoS of the candidate web service $s(i, j)$ of the subtask $t_i$, $p_{i_{max}}$ is the maximum of the property $p$ in the candidate web service set of the subtask $t_i$ and $p_{i_{min}}$ is the minimum. Then $p_{i_{max}}$ is $p_i$ plus several percentage points of ($p_{i_{max}} - p_{i_{min}}$) so that normalized properties are not zero and can be used in Cobb-Douglas function. In this paper, we take 10 percentage points. It may be adjusted according to actual situation.

Then Cobb-Douglas function used as fitness function is:

$$u = nrt\omega_1 na\omega_2 nr\omega_3 nc\omega_4$$

$$\sum_{i=1}^{4} \omega_i = 1 \text{ and } 0 \leq \omega_i \leq 1$$  \(3\)

where $nrt$, $na$, $nr$ and $nc$ represent normalized response time, normalized availability, normalized reliability and normalized cost of composite web services. $\omega_1$, $\omega_2$, $\omega_3$, and $\omega_4$ indicates preference weights of response time, availability, reliability and cost, which are specified by the system or users.

Response time, availability, reliability and cost of composite web services in (3) are calculated based on QoS of component web services as follows:

<table>
<thead>
<tr>
<th>TABLE I. QOS CALCULATION OF COMPOSITE WEB SERVICES</th>
</tr>
</thead>
<tbody>
<tr>
<td>response time</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>$\sum_{i=1}^{n} r_i$</td>
</tr>
<tr>
<td>availability</td>
</tr>
<tr>
<td>reliability</td>
</tr>
<tr>
<td>cost</td>
</tr>
</tbody>
</table>

In Table 1, $m$ represents the number of tasks in a sequence construct, $p$ represents the number of parallel tasks in a parallel construct, $p_i$ represents the probability of the case $i$ in a switch construct where $\sum_{i=1}^{n} p_i = 1$, $k$ represents the estimated number of iterations in a loop construct. The calculation method of QoS and cost of composite web services is adopted from [9, 10].

c) **Genetic manipulation.**

The selection function is the most used roulette selection. And the crossover function is the standard two-points crossover [11] while the mutation function randomly selects a subtask $t_i$ (i.e., a gene in the genome) and then randomly select another web service to replace from $t_i$.

The approach has been used in our previous work [12, 13]. We argue about the advantages of this approach in contrast to GA using the traditional product-addition function in the following section.

III. EXPERIMENT-VALIDITY OF GA USING COBB-DOUGLAS FUNCTION

We use the process combining four kinds of executive logic in Fig. 2 to check the effect of GA using Cobb-Douglas function to search for optimization solutions of web service composition. In Fig.2, ($t_6$, $t_8$) and $t_7$ are parallel tasks, $t_3$ and $t_4$ are switch tasks.

![Figure 2. An example of combination to fulfill a complex task](image)
Experimental tool is MATLAB 7.11.0 (R2010b) Genetic Algorithm Toolbox. We design 5 web services with different characteristics of QoS and cost used for candidate web services for each task. The amount of service composition of this complex task is about $1.95 \times 10^6$. We compare optimization solutions produced by GA using Cobb-Douglas function with that using traditional product-addition function, to check the equilibrium of QoS properties’ contributions to the overall QoS of the web service composition.

First, We compare four optimization solutions of web service composition produced by GA using $u_1 = \prod_{i=1}^4 nq_i^{\omega}$ with those produced by GA using $u_2 = \sum_{i=1}^4 nq_i \times \omega_i$ in the case whether there are preferences in QoS properties.

Fig.3(a) shows the optimization solutions of web service composition produced by GA using the two fitness functions with no preferences for QoS properties, which means there are equal weights on QoS properties. In Fig.3(a), nrt in series 2 is higher than nrt in series 1, and the value of $u_2$ in series 2 is higher than the value of $u_2$ calculated with optimization values of QoS properties in series 1. But the difference between the two values in $u_2$ is a little (0.0847, 4.2% compared to series 2). The contributions of QoS properties to the fitness function in series 1 are more balanced than those in series 2. The conclusion can be intuitively perceived by geometric features (centroid and area).

Fig.3(b) shows the optimization solutions of web service composition produced by GA using the two fitness functions with preferences for QoS properties, which means there are unequal weights on QoS properties. In Fig.3(b), nrt in series 2 is higher than nrt in series 1, and the difference is more in Fig. 3(b) than that in Fig.3(a). It is caused by more weights on nrt. The value of $u_2$ in series 2 is still higher than the value of $u_2$ calculated with optimization values of QoS properties in series 1, and the difference between the two values in $u_2$ is still a little. The contributions of QoS properties to the fitness function in series 1 are more balanced than those in series 2.

The series 2 means users get a solution of service composition which has very low response time, but may be too expensive to be acceptable to users. The series 4 means users get a solution of service composition which has low response time, but is not expensive contrasted with the series 2. The solutions between series 4 and series 2 are similar in overall QoS which are evaluated by the fitness function. So it is better for users to choose the solution of service composition in series 2, which agrees with the logic of making decisions in our daily lives, reflecting customers’ behavior.

In Fig.3(c) and Fig.3(d), we compare the optimization solutions of web service composition produced by GA using the two fitness functions individually from the perspective of different weights on QoS properties.

Fig.3(d) compares the optimization solutions of web service composition produced by GA using the fitness functions $u_1 = \prod_{i=1}^4 nq_i^{\omega}$ in the two weights on QoS properties. In the contrast of the results in Fig.3(c), the
contributions of QoS properties to the fitness function are still balanced. Optimization solutions in series 1 and 3 reflect the characteristics of Cobb-Douglas function aforementioned.

There is a variant of the general product-addition function $u_3 = (n_a \times \omega_a + n_r \times \omega_r + n_{nc} \times \omega_{nc}) / (n_{rt} \times \omega_{rt} + n_{nc} \times \omega_{nc})$. Positive QoS properties such as availability and reliability are in the numerator. The higher they are, the higher $u_3$ is, and the more beneficial for users. On the other hand, negative QoS properties such as response time and cost are in the denominator. The lower they are, the higher $u_3$ is, and the more beneficial for users. All the QoS properties are normalized by (2) for this function.

In the same way, we compare four optimization solutions of web service composition produced by GA using $u_3 = \prod_{i=1}^{m} n_{qi}^{qi}$ with those produced by GA using $u_3$ in the case whether there are preferences in QoS properties.

![Figure 4. Comparison of optimization solutions by GA using the two fitness functions $u_1$ and $u_3$.](image.png)

The results in Fig.4 are the similar with the results in former figures. Optimization solutions in series 1 and 3 reflect the characteristics of Cobb-Douglas function aforementioned. With preferences, the imbalance in optimization QoS solutions produced by GA using $u_3$ is more obvious.

IV. CONCLUSION

This paper proposes an approach of QoS-based web service composition by GA using Cobb-Douglas function. Referring to the theory of customer behavior, the produced service composition not only is high in overall QoS, but also is balanced of QoS properties’ contributions to QoS, which reflects the characteristics of thinking when making decisions in our daily lives. This paper makes an improvement in the fitness function of GA. Other approaches of improved GA used in web service composition, which use the traditional product-addition function, can be replaced by this function to get a better result.

Acknowledgments This paper is sponsored by the National Natural Science Foundation of China under Grant No.61170087 and the Fundamental Research Funds for the Central Universities under Grant No. YWF-13-D2-XX-08.

REFERENCES


