An Approach for QoS-aware Cloud Service Selection Based on Genetic Algorithm and Simplex Method

Chengwen Zhang¹, Jian Kuang², Zhitao Dai³, Bo Cheng⁴, and Lei Zhang⁵
¹Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts & Telecommunications, Beijing, China
²Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts & Telecommunications, Beijing, China
³Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts & Telecommunications, Beijing, China
⁴State Key Lab of Networking and Switching Technology, Beijing University of Posts & Telecommunications, Beijing, China
⁵Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing, China

Abstract - For the problem of cloud service selection, this paper gives a hybrid algorithm. The hybrid algorithm is composed mainly by Simplex Method and Genetic Algorithm. In this algorithm, a tree traversal sequence encoding scheme and some Simplex Method operations are proposed. The design of the tree traversal sequence encoding can support various types of service combinations. In addition, the hybrid algorithm uses Simplex Method operations to further improve local convergence of Genetic Algorithm. The global convergence ability and local convergence capacity of Genetic Algorithm can be gotten better at the same time. Passed tests and analyses show that the algorithm proposed in this paper can be a good choice to solve QoS-based cloud service selection problems.

Keywords: Genetic algorithm, QoS-aware, Simplex method, Cloud service selection

1 Introduction

In cloud computing [1-4] environment, there are inevitably many service providers to provide services with same functionalities and different QoS. These services can combine tens of thousands composite services with same functions and different QoS. That is, there are many different combination plans. Therefore, in a service composition process, we need to choose service components from massive services with same functions and different QoS based on user's QoS requirements. How to select the most suitable composite service among many available candidate services for consumers is an interesting practical issue [24, 25]. The service selection with global QoS constraints possesses a considerably big proportion in the problem of QoS-based cloud service selection. QoS-based cloud service selection plays an important role in the combination of cloud services.

Researches in this area have aroused widespread concern in academic circles [5-11, 14-21, 23-27].

QoS-based cloud service selection problem is one of the hot research areas. The calculation algorithms based on QoS properties is a kind of QoS-based service selection algorithm. Exhaustive methods and approximate algorithms are two kinds of QoS properties calculation methods. To meet the global constraints and to find the optimal combination are under the scope of combinatorial optimization, and QoS-based service selection is NP-hard problem [11], therefore, approximate algorithm is more suitable to solve optimization combinatorial problems. Genetic Algorithm is a kind of approximate algorithm. It is a good method to solve optimization combinatorial problems [12-13]. But, Genetic Algorithm is not advantageous for the local convergence. To compensate for local search capability of Genetic Algorithm itself, the combination of Genetic Algorithm and some kind of local search algorithms is needed to enhance the local search capabilities of Genetic Algorithm.

Based on the above analyses, this paper presents an improved Genetic Algorithm. Firstly, a tree traversal sequence encoding of Genetic Algorithms is described. Secondly, to compensate for the local search capabilities of Genetic Algorithm itself, a hybrid algorithm of Genetic Algorithm and Simplex Method is introduced.

The remaining sections of this paper are as follows. Section two described researches of QoS-based cloud service selection computing. The proposed hybrid GA was discussed in detail in section three. Section four presented some simulation works and discussed the simulation results. Section five came to conclusions and noted that the next step in research content.
2 Quality-based cloud service selection

Based on all global QoS constraints, to select the best plan from a large number of service composition plans is the area of combinatorial optimization. To solve such problems, the calculation methods based on QoS attributes are divided into two categories. One category is an exhaustive algorithm. In this kind of algorithm, all of candidate plans are calculated according to certain rules in order to choose the best plan. It needs to figure out all possible solutions in order to obtain the optimal solution. So the exhaustive combinatorial optimization method is poor scalability and has large calculation. The other is approximate algorithm. In this kind of algorithm, an ideal composition plan is infinitely close to the best one. At last, a plan that meets all QoS requirements but is not the best one will be gained. The methods in [10-11, 16-21] fall into this category.

In the field of combinatorial optimization, there is a random search algorithm based on probability. Genetic Algorithm is suitable for solving such problems [12], and it can effectively prevent exhaustive algorithm limitations. The solution based on Genetic Algorithm is a global optimization one. [10-11, 17-21] used Genetic Algorithm for the optimization of service composition.

Genetic Algorithm and some kind of local search algorithms need to be combined to enhance its local search capabilities and to achieve fairly good results in order to compensate for the local search capability of Genetic Algorithm itself.

3 Genetic algorithm with simplex method

In this section, we present a hybrid algorithm with Simplex Method and GA in order to solve quality-driven selection, mainly including the design of a tree traversal sequence encoding scheme and some Simplex Method operations.

3.1 Tree traversal sequence encoding scheme

A chromosome encoding approach is the basis of the running of Genetic Algorithm. All of subsequent genetic operations should be based on coding design. Different issues should use different coding techniques. The encoding method for solving QoS-based service selection problem should not only reflect every service in services combination but also reflect the structure information of services combination itself. Therefore, the chromosome coding scheme needs to be improved.

This paper designed a tree traversal sequence encoding scheme. The encoding is based on a tree combination template of services combination. The following will in turn introduce the establishment of the tree combination template and how to create chromosomes of GA.

3.1.1 Building tree combination template

A service combination process can be recursively decomposed into four kind of basic models (that is, series model, parallel model, choice model and loop model). Here are four abbreviations SM, PM, CM and LM that express respectively series model, parallel model, choice model and loop model. To describe complicated service combination process, the service combination process can be expressed as an equivalent tree structure that is called tree combination template. In this tree combination template, root node, non-leaf nodes and leaf nodes represent respectively composition model of entire service combination process, logical relationships among tasks and tasks themselves.

![Fig. 1. An example about a service combination process](image1)

Fig. 1. An example about a service combination process

In this example, all of four models are included. In figure 1, T0 and T1, T2 and T3, T4 and T5 are all sequence type. The relationship among tasks T1, T2, T3, T4, T5 and T6 is cycle and its circulation number is m. The two branches of T2 and T4 are selection relationship and their choice probabilities are p1 and p2 respectively. The sum of p1 and p2 is 1. T7 and T8 are parallel type. Figure 2 is the tree combination template that expresses the service combination process in figure 1.

![Fig. 2. A tree combination template corresponding to the service combination process in Figure 1](image2)

Fig. 2. A tree combination template corresponding to the service combination process in Figure 1

In Figure 2, all leaf nodes express all of tasks T0 ~ T8 in the service combination process. Each task can have a number of candidate services. The non-leaf nodes ST, PT, CT and LT express respectively series type, parallel type, choice type and loop type. They are four kinds of expression of their sub-tree relationships. The non-leaf node CT contains the implementation probability of its sub-trees. The non-leaf node LT involves the number of loop of its sub-trees. During
calculating an individual's fitness value, non-leaf nodes will save values of all QoS attributes of its sub-trees.

Before the running of service selection algorithm, it is necessary to express the process of service combination with a unified description. In this paper, a tree-form combination template is used to unify the different formats of services combination processes. After the running of service selection algorithm, the original expression of the services combination processes will be restored.

3.1.2 Creating chromosomes of GA

It is very important to design adequate chromosome to solve the problem of service selection. A tree-encoding is designed in [17]. Each gene in chromosome carries the information of parent nodes and child nodes. In this method, the information of non-leaf nodes was included into chromosome, so that chromosome could carry the logical information of services combination. However, this approach increased the length and complexity of chromosome. Each chromosome must carry same logical information of services combination. This method is a waste of space. And, non-leaf nodes will not participate in follow-up genetic manipulations.

The combinational logic information needs to be kept in tree template and does not need to be included in every chromosome. Chromosome only need include leaf nodes of tree template. The genes of a chromosome arrange in accordance with the result of preorder traversal of tree template. The length of chromosome is equal to the number of tasks in a service combination path. Any additional information does not need to be added into the chromosome. Thus, space can not only be saved, but also the subsequent genetic manipulations are simplified.

According to the characteristics of the four kinds of combination models, it is choice branches to create many service combinatorial paths. The service composition logics until the start point of selection branches are same among some service combinatorial paths. After the start point of selection branches, different service composition logics distribute in different service combinatorial paths. That is, the start point of selection branches is the threshold of different service combinatorial paths.

In the design of chromosome of the hybrid GA, chromosome is established in the form of unidimensional code. All of gene positions in chromosome correspond to tasks in service composition logic. Each task executes in turn successively based on gene position order from left to right. In the relationship between chromosomes and combinatorial logic, different combinatorial path corresponds to different chromosome. That is, the number of chromosomes and the number of combinatorial paths are same.

The creation process of chromosome is as follows.

Firstly, according to the definition of tree combination template, a tree combination template is built based on service composition logic.

Secondly, an unidimensional chromosome is initialized. The leftmost gene locus in the chromosome is the starting point to be treated. A pointer is initialized and points to the leftmost gene locus in the chromosome.

Thirdly, from the root node of the tree combination template, all of nodes are traversed from left to right according to a preorder traversal way.

The tree combination template in Figure 2 can be encoded into chromosomes shown below. In this example, there are two chromosomes that express two composition paths.

T0, T1, T2, T3, T6, T7, T8
T0, T1, T4, T5, T6, T7, T8

After the chromosome has been coded, it could then generate a specific individual (ie, service combination instance). The following is the specific method to build an individual. Every gene in the chromosome expresses a task of service composition. There are a lot of candidate services for a task. A specific atomic service needs to be selected from every set of candidate services. All of atomic services will form a composite service instance that is an individual in population. A certain number of generated individuals can compose an initial population. Through main genetic manipulations, new generations of population constantly are gotten until a satisfactory solution is obtained.

3.2 SM operations

GA can not be restricted by restrictive assumptions constraints in search space. It does not require continuity, derivative existence and single peak assumptions. In addition, Genetic Algorithm with global optimization capability uses populations to organize optimization operations. It searches multiple regions in the solution space at same time, so it has the inherent parallelism. However, local convergence of Genetic Algorithms is not an advantage. Therefore, in order to compensate for Genetic Algorithm itself in lack of local search capability, Genetic Algorithm needs to be integrated with some kind of local search algorithms to enhance its local search capabilities.

Simplex Method (SM) is a common approach to solve mathematical programming problem. Genetic Algorithm and Simplex Method also have their own advantages. Genetic Algorithm has global optimization capability, and it can search simultaneously multiple regions of solution space. Simplex Method has local space search ability, fast
convergence speed. It can change search direction according to the trend of fitness values and the use of local information.

Existing researches primarily solve the problem of combination service selection through local optimization or global optimization algorithms. The running time of the local optimization algorithm is less than the global optimization algorithm, but it can not take into account the global QoS constraints, and often can not meet non-functional requirements that customers bring forward. The global optimization algorithms can consider the global QoS constraints, but the amount of calculation is larger than the local optimization algorithm. Thus, these two algorithms both have some limitations.

As described above, a combination of Genetic Algorithm and Simplex Method can form a hybrid algorithm [22] that includes the global optimization algorithm and the local optimization algorithm. Genetic Algorithm ensures that the hybrid algorithm has the global search capability and can find the global optimal point. Simplex Method can add a number of parallel searches in many local areas and it can use local search methods to direct the search. It can not only speed up the process of global optimization, but also solve the "premature" problem of Genetic Algorithm to a certain extent. Better convergence speed and search capability can be gotten at the same time.

Based on the research about the combination of simplex method and Genetic Algorithm, this paper presents a hybrid algorithm that is the Combination of Genetic Algorithm and Simplex Method. This hybrid algorithm will be used to solve the service choice problem.

The following is the main idea of the hybrid algorithm. After Genetic Algorithm produces a new generation of population, some local initial simplexes are composed by some randomly selected individuals in a certain probability. Individuals with higher fitness values are introduced through continuous reflection operations and they will replace the individuals whose fitness values are lower. So, a number of new better individuals will be included into the next generation of population. In addition, during the reflection operation, the decision variable matrix will be used.

Some Simplex Method operations are joined between two generations of population. After a series of reproduction, crossover and mutation operators, a number of individuals are randomly selected to form a certain number of initial simplexes. Some local Simplex operations are run in parallel. After all initial simplexes have completed their simplex operations, more excellent individuals are obtained. We can proceed with the next generation of genetic manipulations.

\[ N_{is} = \text{ceil} \left( \frac{Sp}{N_{task} + 1} \right) \]  

In the above equation, The number of generated initial simplexes is \( N_{is} \). \( Sp \) is the population size of Genetic Algorithm. \( N_{task} \) is the number of tasks.

The main steps of simplex operations of each initial simplex are the following ones:

1), Establishing an initial simplex

An initial simplex is formed in a n-dimensional space by \( n+1 \) individuals that are selected randomly from the current population.

2), Selecting the worst individual

The vertex with the smallest function value among \( n+1 \) vertices is found and its corresponding individual is denoted by \( I_{n+1} \). The individuals corresponding to the remaining \( n \) vertices are indicated respectively by \( I_1, I_2, \cdots, I_n \).

3), Constructing decision variable matrix of every vertex

The decision variable matrices \( D_1, D_2, \cdots, D_n, D_{n+1} \) are built respectively for individuals \( I_1, I_2, \cdots, I_n, I_{n+1} \). In the decision variable matrix, each row represents the decision variable vector of all candidate services of a task. As shown below is the specific method to construct. Only when the \( j \)th candidate service of the \( i \)th task is selected, the component \( d_{ij} \) is 1 in the decision variable matrix \( D_k \), otherwise the value of \( d_{ij} \) is 0.

4), Calculating decision variable matrix of reflection center

The reflection center is \( I_c \) that is about \( n \) individuals except the worst individual \( I_{n+1} \). The decision variable matrix \( D_c \) about \( I_c \) can be built according to the following equation.

\[ D_c = \left( \sum_{i=1}^{n} D_i \right) / n \]  

5), Computing decision variable matrix about the reflection point

\( I_0 \) is the reflection point of the worst individual \( I_{n+1} \) on \( I_c \). Its decision variable matrix is \( D_0 \).
6), Boolean the decision variable matrix of the reflection point

Boolean-oriented approach is to reassign 0 or 1 to each component \( d_{ij} \) in the decision variable matrix \( D_0 \). The value 1 will be set to the largest component in each row vector \( D_k \) of \( D_0 \) and the remaining components are assigned the value of 0. Thereby, a boolean decision variables matrix \( D^{0'} \) will be generated. In the Boolean process, if there are multiple components with the same and maximum value in a row vector, the value 1 will be set to random component among them. The remaining components in the row vector are 0.

7), Generating a new individual corresponding to the reflection point

A new individual \( I_0 \) is generated on the basis of the decision variable matrix \( I_0^{0'} \). For each row vector in \( D^{0'} \), the only component with the value of 1 is used to select its atom service instance. The atom service instance will be assigned to corresponding gene locus on a chromosome. After all of gene loci are set atom service instances, the formation of a new individual \( I_0 \) will be done.

8), Determining whether new individuals meet user's global constraints

Based on the search thinking of that the best point should be almost the opposite of the worst one, the fitness value of a new individual \( I_0 \) is always greater than the worst individual \( I_{n+1} \) in the initial simplex. Therefore, if the new individual's fitness is greater than the worst individual and the new individual meets the user's global QoS constraints, the new individual will replace the worst one in population and joins the next generation population evolution. Otherwise, if the new individual's fitness is less than the worst individual or the new individual does not meet the user's global QoS constraints, the new individual will also replace the worst one in population and form a new simplex to continue with the next iteration of the simplex algorithm. We can end the operation of the simplex until a new individual's fitness is greater than the worst individual and the new individual meets the user's global QoS constraints.

Simplex operations are done in \( N_{ij} \) initial simplexes in turn. After every simplex has gained a new individual whose fitness value is better than the worst individual in the simplex and that is able to meet the global user constraints, these new individuals will be generated and added into the population to participate in the next generation of population genetic manipulations.

Because individuals are randomly selected to build an initial simplex, the randomness of Genetic Algorithm can be ensured. And the opportunities to generate new individuals are increased. On the other hand, Simplex Method can control the evolution direction of Genetic Algorithm to make better solutions. It is parallel searches in a number of local solution spaces not only that enhances the local search ability but also that accelerates the global convergence and solves the "premature" problem of GA to a certain extent.

4  Tests and analyses of hybrid GA

The proposed service selecting algorithm in this paper improves simple Genetic Algorithm in two ways. One is the improvements of simple Genetic Algorithm including tree traversal sequence encoding. The other hand is to build a more powerful and efficient hybrid search algorithm that is composed by Genetic Algorithm and Simplex Method. Through the above two improvements, the hybrid GA has better search ability. Here are the tests and test analyses through which the capacity of the presented hybrid GA will be validated.

4.1  Test data preparation

In order to verify the effect of services choice done by the hybrid GA, some comparison tests between simple Genetic Algorithm and the hybrid GA algorithm were made.

In order to fairly test the two algorithms, they would run in the same hardware and software operating environment, including CPU, memory, OS, development language and IDE, etc.

The simple Genetic Algorithm and the hybrid GA used initialization parameters as follows. The population size is 500. The crossover probability is 0.7 and the mutation probability is 0.1.

Based on the above preparation of test data, simple Genetic Algorithm and the hybrid GA were run respectively. The test results were analyzed from search capability

4.2  Tests and analyses of search capability

Search capability is that the algorithm can find the optimal solution in a solution space. It can be measured by the quality of the solution that the algorithm searches. In Genetic Algorithm, the algorithm search capability can be measured through the fitness value of the final selected individual.

The hybrid GA took a tree traversal coding as well as the combination of Simplex Method to improve global search ability and local search capability from different aspects of Genetic Algorithm. In order to verify these strategies, simple Genetic Algorithm and the hybrid GA were run for 50 times at different scale of problems (that is, the number of different
tasks and different number of candidate services) respectively. The average values of the final fitness values at all running time were taken. A few set of test data are listed in Table I.

<table>
<thead>
<tr>
<th>task number</th>
<th>average number of candidate services for each task</th>
<th>average fitness value of simple GA</th>
<th>average fitness value of hybrid GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>0.227</td>
<td>0.239</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>0.106</td>
<td>0.197</td>
</tr>
<tr>
<td>25</td>
<td>35</td>
<td>0.027</td>
<td>0.09</td>
</tr>
</tbody>
</table>

As shown in Table 1, when in the face of the selection problem with the same size of combination services, the hybrid GA can get higher average final fitness value than the simple Genetic Algorithm. When the scale of the composition problem is small, the advantage of the hybrid GA is not clear. But, when there are a larger number of tasks in a combined service flow, the hybrid GA can get much better solutions than the simple Genetic Algorithm. In the test conditions of this article, when the number of tasks is more than 12, the hybrid GA clearly has stronger search capabilities. This shows that the hybrid GA has better search capabilities, especially in the larger scale of service selection, that the search capabilities are more prominent.

The crossover operation of chromosomes and the mutation operation of chromosomes can search optimal results not only in same path but also in different composition paths. The search areas of the crossover operation and the mutation operation are broadened. This is one reason why the hybrid GA can get more fitness than the simple GA. Another reason is the use of Simplex Method operations. These Simplex Method operations enhance the local search capabilities of Genetic Algorithm.

5 Conclusions

As a kind of distributed computing model, academia and industry have been greatly concerned about cloud services in recent years. With the cloud services technologies have become more sophisticated, more and more easily used cloud services with the stability characteristics are shared on network. But a single atomic cloud services can provide limited functionalities. In order to more fully utilize the shared cloud services, it is necessary to combine shared cloud services to form a new combination of cloud services to provide more powerful service functions.

With the progressive development of cloud services technology and application, it is inevitable for a task to appear a large number of candidate services with the same function properties and different non-functional attributes (mainly referring to QoS attributes). It has become an urgent problem that how to fast and flexibly select a high-availability, high reliability, high performance and the best services to meet user’ needs from massive candidate services. Namely, it is QoS-based service selection problem.

This paper presents a combination services selection algorithm based on the hybrid GA. Based on the analyses of the 0-1 integer programming model of composite service selection problem, the simple Genetic Algorithm is improved itself and combines a local optimization algorithm – Simplex Method. The improvements of the simple Genetic Algorithm itself includes that the design of a tree traversal sequence encoding to support a variety of combination types. In addition, in order to compensate the lack of the ability of local search of Genetic Algorithm itself, Genetic Algorithm and Simplex Method are applied to the formation of a new hybrid algorithm. In the result, the search ability can be improved at the same time.

Through the realization of the above-mentioned algorithm, testing and analyses of test results, some strong validations of the proposed algorithm in capacity effect were done. The hybrid GA can be a good solution to QoS-driven cloud services selection.

In the above experiments, the number of individuals in populations is same in the face of different combination sizes. If the populations with different sizes can be adopted for different composition scales, the efficiency of algorithm will be greatly improved. Therefore, the next study will examine the dynamic adaptive mechanism of population size. The other next step is to apply the proposed hybrid algorithm into a number of practical large-scale services computing environments, in order to improve the efficient and reliable operations of the hybrid GA further.

Acknowledgments: This work was supported by NSFC under Grant Nos. 60872042, the Fundamental Research Funds for the Central Universities (2011RC0203) and the Co-construction Special Funds of Beijing.

6 References


