A Hybrid Algorithm Based on Genetic Algorithm and Simplex Method for QoS-aware Cloud Service Selection

Chengwen Zhang¹, Jiali Bian², Bo Cheng³, and Lei Zhang⁴

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
³State Key Lab of Networking and Switching Technology, Beijing University of Posts & Telecommunications, Beijing, China
⁴Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing, China

Abstract - Aiming at the problem of cloud service selection based on global Quality-of-Service (QoS) constraints, this paper provides a hybrid algorithm of Simplex Method (SM) and Genetic Algorithm (GA). In this algorithm, some relevant variables are defined, some Simplex Method operations are proposed and the hybrid algorithm based on GA and SM is provided. The global convergence ability and local convergence capacity of GA can be gotten better. The hybrid algorithm can get more excellent composite service plans from a lot of composite plans on the basis of global QoS constraints because it accords with the characteristics of cloud service selection very well. Passed tests and analyses show that the hybrid algorithm proposed in this paper can be a good choice to solve the QoS-based cloud service selection problem.

Keywords: Cloud service, Selection, QoS-aware, Genetic algorithms, Simplex method

1 Introduction

Cloud computing [1-3, 31] is an emerging computing technology that allows businesses to implement their own services using on-demand IT infrastructures. These on-demand infrastructures enable end users to access business services without installation, at any computer with Internet access. Cloud computing environments offer three major types of services: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS)[4]. These services are called cloud services. Cloud service composition is one of the motive forces of the development of cloud services.

With the rapid development of cloud service technology, as well as intensified competition among cloud service providers, there are inevitably many cloud service providers to provide cloud services with same functionalities and different QoS. These cloud services can combine tens of thousands composite cloud services with same functions and different QoS. Therefore, we need to choose cloud service components from massive cloud services with same functions and different QoS based on user's QoS requirements. QoS-based service selection plays an important role[5-6] in the combination of services.

QoS-based service selection problem is one of the hot research areas. A lot of international research organizations in this field carried out relevant research work and have made some research results [7-25, 29-30]. But there are still certain deficiencies.

Exhaustive methods [7-9, 12, 20-25] and approximate algorithms [10-11, 13-19] 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 (GA) is a kind of approximate algorithm. Genetic Algorithm is a powerful tool to solve combinatorial optimizing problems [26]. It is an iterative procedure on the basis of population where each individual describes a solution. The design of Genetic Algorithm operators and parameters will have significant impact on itself [27]. Genetic Algorithm is not advantageous for the local convergence. Its efficiency is not enough and its speed of convergence is slow. In order 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 analysis, this paper presents an improved Genetic Algorithm. To compensate for the local search capabilities of Genetic Algorithm itself, a hybrid algorithm of Genetic Algorithm and Simplex Method (SM) 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 algorithm 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 cloud service composition plans is in the area of combinatorial optimization. To solve such problems, the calculation methods based on QoS attributes are divided into two categories. One category is exhaustive algorithm. In this kind of algorithm, all of candidate plans are calculated according to certain rules in order to choose the best plan. The methods used in the literatures [7-9, 12, 20-25] fall into this category. The other is approximate algorithm. In this type 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 the literatures [10-11, 13-19] fall into this category.

QoS properties calculation through the establishment of QoS matrix is a representative calculation method. The literature [7] presented a run-time services choice method in dynamic service composition. It could select a single better service, but it could not meet the entire QoS requirements. In the literatures [8, 9], a local optimization algorithm and a global optimization algorithm were proposed. The local optimization algorithm could not reach a global optimal solution. When the size of composition services was large, the computation of the global computational algorithm had increased a lot. The literature [12] expanded the methods in the literature[9]. It analysed the conditions of triggering service re-selection in detail, gave the idea of the service re-selection and gave the constraint expression for a stateful cloud service selection.

The service selection problem based on QoS belongs to NP-hard problem[11], so the exhaustive combinatorial optimization method is poor scalability and has large calculation. Heuristics method can be used to obtain an approximate solution.

In the literature [13], a multidimensional 0-1 knapsack problem model was used for multiple QoS constraints selection. A method based on branch-and-branch was proposed for solving MMKP (Multi-dimension Multi-choice Knapsack Problem) optimal solution and heuristic-based method for solving second-best solution. In the field of combinatorial optimization, the solution based on Genetic Algorithm is a novel global optimization one. The literatures [10-11, 14-18] used Genetic Algorithm for the optimization of service composition. The literature [10] used Genetic Algorithm to solve the QoS-based service selection. It used one-dimensional chromosome encoding method to describe the combination of services. The literature [11] also used an one-dimensional chromosome encoding method to describe the combination of services. The literatures [17] and [18] proposed a combination service method based on Genetic Algorithm. Through Genetic Algorithm, it could be ensured that the results of services choice met the restrictive conditions.

To compensate for the local search capability of Genetic Algorithm itself, 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.

3 Genetic algorithm with simplex method

In this section, we present a novel genetic algorithm with Simplex Method in order to solve quality-driven selection.

3.1 Definition of relevant variables

Suppose that a composite cloud service includes n tasks. The i-th (1 ≤ i ≤ n) task t_i has N_i candidate services. The sign s_{ij} is used to represent the j-th (1 ≤ j ≤ N_i) candidate cloud service. The sign Q_{ijq} denotes the q-th QoS attribute of s_{ij} candidate cloud service. The weight value of Q_{ijq} is W_{q}.

The symbol y_{ij} denotes a decision variable. In a composite cloud service instance, the value of y_{ij} is 1 only when the cloud service s_{ij} is selected, otherwise its value is 0. For the task t_i, only one decision variable value is 1, the rest values are 0. Its formula is the following formula (1).

$$\sum_{j=1}^{N_i} y_{ij} = 1 \cdot y_{ij} \in [0, 1]$$

In addition, in normal circumstances, the value of N_i of every task is usually not equal one another. Suppose that m is the maximum value in all of N_i, namely, m = Max [N_1, N_2, ..., N_n]. Therefore, in order to build a decision variable matrix formed by all decision variables, the number of candidate cloud services of every task needs to be expanded to m. The following is the method. If N_i < m for a task i (1 ≤ i ≤ n), the value of y_{ij} is 0 in the case of N_i < 1 ≤ j ≤ m. Accordingly, the total number of the decision variables is expanded to n × m. Since the new expanded candidate cloud services will never be selected, the corresponding decision variable values are always 0.

Based on the above expansion, all of decision variables y_{ij} (1 ≤ i ≤ n, 1 ≤ j ≤ m) can constitute a n×m class of decision variable matrix that is denoted by Y. Its formula is the following formula (2).
In Y, each row represents the decision variable vector of all candidate cloud services of a task.

3.2 Main steps of the simplex operations

Genetic Algorithm can effectively handle the optimization problem with multi-variable and complex functions. 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 is a local optimization approach. A combination of Genetic Algorithm and Simplex Method can form a hybrid algorithm [28] that includes the global optimization algorithm and the local optimization algorithm. The two algorithms complement each other. 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 cloud 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 and will participate in genetic manipulations in 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.

\[ N_s = \text{floor}(N_g / n) \]  

(3)

In the formula (3), \( N_g \) is the population size of Genetic Algorithm, that is, it is the number of individuals in each generation of population. The symbol \( n \) is the length of a chromosome.

For each initial simplex, the main steps of the simplex operations are as follows:

3.2.1 To establish an initial simplex

\( n+1 \) individuals are selected randomly from the current population and form an initial simplex in a \( n \)-dimensional space. Each individual's fitness function value shall be the function value of the corresponding vertex in the simplex.

3.2.2 To select the worst individual

After the function values of \( n+1 \) vertices are compared, the vertex with the smallest function value is found and its corresponding individual is denoted by \( C_1 \). The individuals corresponding to the remaining \( n \) vertices are indicated respectively by \( C_2, C_3, \cdots, C_n \).

3.2.3 To construct the decision variable matrix of every vertex

The decision variable matrixes \( Y_1, Y_2, \cdots, Y_n, Y_{n+1} \) are built respectively for the individuals \( C_1, C_2, \cdots, C_n, C_{n+1} \). As shown below is the specific method of construction. Only when the \( j \)th candidate cloud service of the \( i \)th task is selected, the component \( y_{ij} \) is 1 in the decision variable matrix \( Y_i \), otherwise the value of \( y_{ij} \) is 0.

3.2.4 To calculate the decision variable matrix of reflection center

\( C_c \) is the reflection center that is about \( n \) individuals except the worst individual \( C_{n+1} \). The decision variable matrix \( Y_c \) about \( C_c \) can be built according to the following formula (4).
\[ Y_c = \frac{1}{n} \left( \sum_{k=1}^{n} Y_k \right) \] (4)

3.2.5 To compute the decision variable matrix about the reflection point

\( C_0 \) is the reflection point of the worst individual \( C_{n+1} \) on \( C_r \). Its decision variable matrix is \( Y_0 \). Its formula is the following formula (5).

\[ Y_0 = 2 Y_c \cdot Y_{n+1} \] (5)

3.2.6 Boolean the decision variable matrix of the reflection point

Boolean-oriented approach is to reassign 0 or 1 to each component \( Y_{ij} \) in the decision variable matrix \( Y_0 \). The value 1 will be set to the largest component in each row vector \( Y_k \) of \( Y_0 \) and the remaining components are assigned the value of 0. Thereby, a boolean decision variables matrix \( Y_0^* \) 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.

3.2.7 To generate the new individual corresponding to the reflection point

A new individual \( C_0 \) is generated on the basis of the decision variable matrix \( Y_0^* \). For each row vector in \( Y_0^* \), 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 \( C_0 \) will be done.

3.2.8 To determine whether the new individuals meet the user's global constraints

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.

In accordance with the above steps, simplex operations are done in \( N_s \) 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.

On the one hand, because individuals are randomly selected to compose 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 bus also that accelerates the global convergence and solves the "premature" problem of GA to a certain extent.

3.3 GA with SM

In this section, we present a novel genetic algorithm mixed by SM. This GA is available in figure 1.

![Fig.1. GA with SM operations](image)

The SM operations promote the search ability of GA. The hybrid GA can gain better composition services.
4 Tests and analyses

The proposed cloud service selecting algorithm in this paper improves simple Genetic Algorithm. That is to build a more powerful and efficient hybrid search algorithm that is composed by Genetic Algorithm and Simplex Method. Through the above improvement, this algorithm has better search ability. Here are tests results and test analyses through which the capacity and efficiency of presented hybrid algorithm will be validated.

4.1 Test data preparation

In order to verify the effect of cloud services choice done by the hybrid algorithm, some comparison tests between simple Genetic Algorithm and the hybrid 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. In the comparison tests, the two algorithms solved the cloud service selection problems with the same size of cloud services combination. In every comparison test, the number of tasks in cloud service composition was same. The values of specific QoS attributes were randomly generated within a certain range.

Some global limits for a part of QoS properties were randomly generated. The overall limits were applied to all specific cloud service compositions through the penalty function method.

In addition, the simple Genetic Algorithm and the hybrid algorithm used initialization parameters as following. The population size is 500. The crossover probability is 0.7 and the mutation probability is 0.1.

4.2 Tests and analyses

Based on the above preparation of test data, simple Genetic Algorithm and the hybrid algorithm were run for 50 times respectively at different scale of problems (that is, the number of different tasks and different number of candidate cloud services). The test results were analyzed from both convergence speed and search capabilities.

Algorithm convergence rate refers to the generation where the biggest fitness value is reached. The average running time was taken too. 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. In the hybrid algorithm, the bigger the fitness value is, the better the selection result is.

The average values of the final fitness values at all running time were taken.

A few of test data are listed in Table 1. SGA is the abbreviation of simple Genetic Algorithm.

<table>
<thead>
<tr>
<th>Tasks Num</th>
<th>Average</th>
<th>Maximum</th>
<th>Average</th>
<th>Average Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.134:0.13</td>
<td>9</td>
<td>256:537</td>
<td>343:284</td>
</tr>
<tr>
<td>25</td>
<td>0.061:0.17</td>
<td>3</td>
<td>983:2135</td>
<td>386:297</td>
</tr>
<tr>
<td>30</td>
<td>0.031:0.10</td>
<td>5</td>
<td>2676:537</td>
<td>352:308</td>
</tr>
</tbody>
</table>

As can be seen from Table 1, when the number of tasks is added, the running time will increase.

The efficiency of simple Genetic Algorithm is still unsatisfactory, although to a certain extent it solved the cloud service selection problem. As described in Table 1, comparison of data can fully verify that the hybrid algorithm has faster convergence than the simple Genetic Algorithm and can get better results of cloud service selection.

When in the face of the selection problem with the same size of combination cloud services, the hybrid algorithm 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 algorithm is not clear. But, when there are a larger number of tasks in a combined service flow, the hybrid algorithm 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 25, the hybrid algorithm clearly has stronger search capabilities. This shows that the hybrid algorithm has better search capabilities, especially in the larger scale of cloud service selection, that the search capabilities are more prominent. The reason is that the presented hybrid algorithm in this paper has greatly enhanced the local convergence rate and capability with the combination of Simplex Method and Genetic Algorithm.

But, the average running time of the hybrid algorithm were larger than the SGA. This means that the running of Simplex Method raises the running time of the hybrid algorithm. Some improvement methods should be established in the future.

5 Conclusions

Since 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 service 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 cloud services with the same function properties and different non-functional attributes (mainly referring to QoS attributes). It has become an urgent problem how to fast and flexibly select a high-availability, high reliability, high performance and the best cloud service to meet user’ needs from massive candidate cloud services. Namely, it is QoS-based cloud service selection problem.

This paper presents a combination cloud services selection algorithm based on a hybrid algorithm. Based on the analyses of composite cloud service selection problem, a simple Genetic Algorithm combines a local optimization algorithm – Simplex Method. 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 and convergence speed can be improved at the same time.

Through the realization of the above-mentioned algorithm, some strong validations of the proposed algorithm in capacity and efficiency effects were done. The hybrid algorithm can be a good solution to Qos-driven cloud services selection.

In the future, some ways should be adopted to decrease the running time of the hybrid algorithm. In the above experiments, the number of individuals in populations is the 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. An approach of Genetic Algorithm with ant colony algorithm is used to select concrete services in the literature 19. MMAS (Max-Min Ant System) has abilities of parallel processing and global searching. MMAS can put feedback information to Genetic Algorithm. So, another future work is about how Genetic Algorithm works together with MMAS in dynamic cloud service selection environment.

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


