

Web service composition based on improved multi population genetic algorithm

Siyuan Meng

School of Computer Science of Technology, Shandong University of Technology, Zibo, 255000, China

mengsiyuan@gmail.com

Keywords: Web service composition, multi-population genetic algorithm, QoS, big mutation operator.

Abstract: With the development of cloud computing, the improvement of web service standards and the progress of supporting software, more and more web services are published on the Internet. Web service quality aware (QoS) not only requires specific services to complete specific tasks, but also pays more attention to the comprehensive service quality of the whole web service composition. How to select the web service composition with the highest comprehensive QoS in the global is NP hard. In this paper, an improved two population genetic algorithm is proposed, in which a adaptive crossover operator is set in one population and a big mutation operator is set in another population to improve the existing genetic algorithm, so that the algorithm can balance the local search and global search ability. The experimental results show that this algorithm has the advantages of shorter time-consuming and higher accuracy than the general genetic algorithm and the multi-population genetic algorithm, and effectively avoids the defect of effective genes in the population.

1. Introduction

1.1 Research Significance and Background

In recent years, with the continuous development of cloud computing, more and more web services are published on the web. Individual services can no longer meet the needs of users, and more and more users are composing Web services to meet various business processes to achieve more complex functions. The powerful service function of Web service composition can save costs and create greater benefits for enterprises. Therefore, the appropriate service composition in the selection of large number of Web services in the Internet has become the focus of current e-commerce research.

Web service composition is the process of selecting specific services from several abstract tasks and combining them into large granular services. From the perspective of business process, reference [1] thinks that web composition service is to connect services organically according to some business rules, and that each abstract service in service composition cooperates with each other to achieve specific goals. From the aspect of solving web service composition, reference [2] thinks that web service composition is a process of finding a service composition that can achieve the user's specific

goal in a given series of services。 For a web service composition, each abstract task can be completed by different specific services. Each specific service has similar functions, but users can filter the service through a series of non-functional attributes: the QoS attribute is used to constrain the service. QoS attributes are used to measure the quality of each service. However, in the case of meeting the local QoS constraints, it is unable to guarantee the global optimal QoS of the whole web service composition. Therefore, how to find a global QoS optimal solution under the premise of meeting the local constraints and completing the task process has become a problem to be solved.

1.2 Research at Home and Abroad

With the continuous development of intelligent optimization algorithms in recent years, more and more swarm intelligence optimization algorithms are studied and applied in the field of web service composition. For example, the cuckoo search based web service composition method proposed in reference [3] and the ant colony algorithm based web service composition model proposed in reference [4], but at the same time, these methods cannot avoid falling into local optimum. Literature [5] proposed a web service composition model based on particle swarm optimization, and literature [6] proposed a web service composition model based on genetic algorithm. Compared with the traditional optimization algorithms including weighting method, constraint method and linear programming method, a large number of studies show that genetic algorithm has better advantages for large-scale web service composition problems. However, genetic algorithm has the disadvantages of slow astringency and easy to fall into the local optimal solution. In recent years, a large number of scholars have improved the genetic algorithm from different perspectives. Reference [7] gives the application of genetic algorithm in QoS aware web 'service composition. In reference [8], it is proposed for the first time to code genes in the form of matrix, which effectively reflects the relationship between services and solves the problem of multi-path using a coding representation. This series of improved algorithms, in varying degrees, promote genetic algorithm to find a better solution in the service composition problem.

In view of the above problems, this paper proposes an improved double population genetic algorithm based on literature [9] to solve the problem of web service composition. By setting big crossover operator and mutation operator in two different populations, the global and local search ability of genetic algorithm is improved. The simulation results show that the algorithm has high superiority and feasibility, and can meet the actual needs.

2. Background and Knowledge

2.1 Web Services Composition Based on QoS

Definition 1 quality of service refers to the collection of network service requirements of business flow in the process of network transmission.

Among them, traffic flow refers to the packet flow related to specific QoS and from source to destination. Therefore, QoS is considered as a set of non-functional requirements for measurable web services, which generally include time, cost, availability, and reputation.

Definition 2 generally we can pass a four tuple (S, C_{in}, C_{out}, Q) . S represents the finite set of services, C_{in} represents the finite set of input services, C_{out} represents the finite set of output services, and Q represents the finite set of QoS.

In the process of business flow oriented web service composition, the connection modes of each abstract service can be divided into parallel, sequential, and selected modes, such as (w_1, w_2, \dots, w_n) is connected in a sequential manner, and $(w_1 || w_2 || \dots || w_n)$ from w_1 to w_n is executed in parallel. According to the above definition, we give the following table 1 to calculate the QoS calculation

formula of Web services with different connection model.

Table 1: QoS calculation formula of composite service

LINKE MODE	COST	TIME	AVA	REP
SEQUENCE	$C = \sum_{i=1}^n C_i$	$T = \sum_{i=1}^n T_i$	$A = \prod_{i=1}^n A_i$	$R = \sum_{i=1}^n R_i/n$
PARALLEL	$C = \sum_{i=1}^n C_i$	$T = \max(T_1, T_2, \dots, T_n)$	$A = \prod_{i=1}^n A_i$	$R = \sum_{i=1}^n R_i/n$
SELECTION	$C = \sum_{i=1}^n a_i C_i$	$T = \sum_{i=1}^n a_i T_i$	$A = \sum_{i=1}^n a_i A_i$	$R = \sum_{i=1}^n a_i R_i$

In the process of QoS calculation, due to the different measurement units of response time, availability and other attributes, the range of values is different, which will lead to the proportion of some attributes being too large, which will affect the overall results. So we normalize QoS by formula (1) and (2) and convert them to values of 0 to 1. For the negative attributes of response time and price, we normalize the attributes by (1), and normalize the attributes by formula (2) for positive attributes such as reputation, availability, throughput, etc.

$$f(x) = \begin{cases} \frac{Q_i^{max} - Q_i(w_j)}{Q_{max} - Q_{min}}, & \text{if } Q_{max} - Q_{min} \neq 0 \\ 1, & \text{if } Q_{max} - Q_{min} = 0 \end{cases} \quad (1)$$

$$f(x) = \begin{cases} \frac{Q_i(w_j) - Q_{min}}{Q_{max} - Q_{min}}, & \text{if } Q_{max} - Q_{min} \neq 0 \\ 1, & \text{if } Q_{max} - Q_{min} = 0 \end{cases} \quad (2)$$

2.2 Flow of Genetic Algorithm

Genetic algorithm is a random search algorithm based on probability proposed by Professor John Holland and his students of the University of Michigan in the 1970s. Genetic algorithm encodes the genes of chromosomes in the initial population, and each locus is combined to form a chromosome, which represents the individual's phenotype. After the formation of the primary population, the algorithm simulates the genetic law in nature, and selects the excellent individuals by setting the adaptive function to implement the rule of survival of the fittest. Then the genetic operator in genetics is introduced to operate crossover and mutation, so as to obtain the next generation population. In this process, the new generation population is more adaptive to the environment than the previous generation. After continuous iteration, the last generation population can be approximately regarded as the optimal solution which is most suitable for the current environment.

(1) First generation population is generated, and a certain number of individuals are randomly generated, which are expressed as gene codes on chromosomes.

(2) The fitness function is used to calculate the fitness of each individual. If the optimal solution is found, the calculation ends, otherwise, the execution continues.

(3) In order to select good individuals and reduce the mating probability of bad individuals..

(4) Perform crossover and mutation operations to generate new populations

(5) After obtaining the new generation population, return to the second step

But at the same time, the traditional genetic algorithm has the problem of premature convergence,

the population is easy to lose the genetic diversity in the early stage, and fall into the problem of local search. In addition, when the amount of data is too large, or the number of individuals in the population is not enough, the probability of excellent genes will be reduced, so the optimal solution cannot be obtained. For example, if there are too many specific services in an abstract service, fewer individuals and low mutation rate will reduce the possibility of optimal gene. Based on the above problems, this paper proposes a web service composition method based on double population genetic algorithm to solve the problem, which can make the algorithm jump out of the local extreme point and create a new search plane, effectively avoiding the premature of the algorithm.

3. Algorithm Design and Implement

3.1 The Expression of Chromosome

In the algorithm proposed in this paper, a chromosome represents a group of service composition, an abstract service represents a locus on the chromosome, the number of abstract services is the number of loci, and each specific service can be represented as a gene. We put the specific services with similar functions into a service set to generate a candidate service set. Each locus will correspond to a specific candidate service set, and a specific service in the candidate service set will be selected to complete a task in the business process. Selecting a service on each locus to form a chromosome, which represent a web service composition.

In terms of chromosome structure and gene coding, the integer coding method used in this paper is similar to that in reference [7]. A chromosome structure contains n loci, and each locus contains the connection type of abstract service, the subscript index of specific service in the service candidate set, and the index to measure the service quality calculated by the fitness function in formula (3). On each gene of the locus, the array stored the QoS attributes of each current service, including time, cost, reputation and availability. The specific chromosome structure is shown in figure (1).

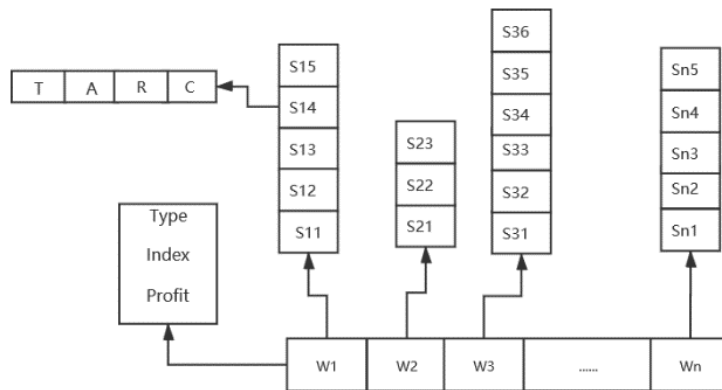


Figure 1: The chromosome structure

3.2 Implement of Improved Multi-Genetic Algorithm

In the process of genetic algorithm convergence, with the continuous iteration, the individuals in the population become more and more single, and the diversity of the population decreases, so it will fall into the local optimal solution. Most of the reasons for this situation are due to the lack of excellent genes in the population, which leads to the algorithm only looking for the optimal solution in some local candidate services. In order to overcome the above problems, this paper proposes a model based on the combination of parallel multi population structure and elite multi population structure. Figure (2) shows the model of this algorithm.

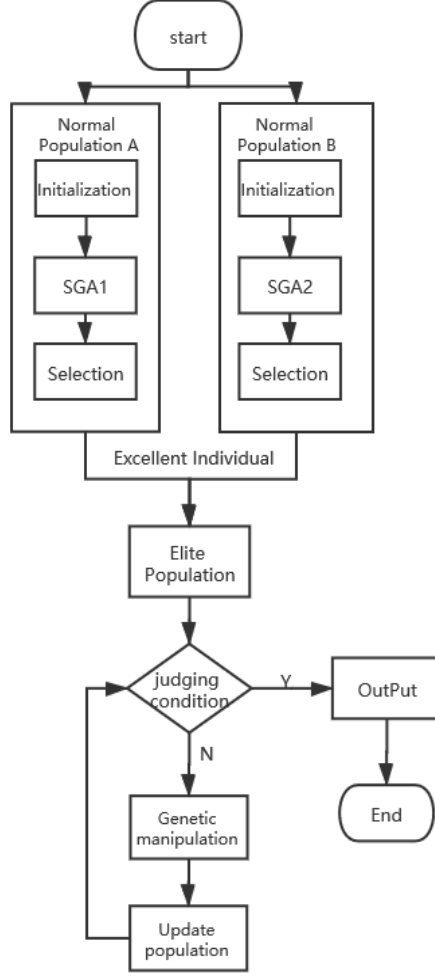


Figure 2: Algorithm flow chart

After initializing the populations respectively, the two populations execute sga1 algorithm of high crossover operator and sga2 algorithm of high mutation operator respectively, and the excellent individuals of each generation are selected and put into the elite population. Then the genetic algorithm is executed again from the elite population, and the optimal individuals are obtained by the final results.

In genetic operation, we use the single point crossing method to minimize the possibility of reducing individual fitness. The specific operation is as follows: C_1, C_2 is the chromosome operated by the parent, C'_1, C'_2 is the chromosome after the cross operation and ∂ is the cross part.

$$\begin{cases} C'_1 = \partial C_1 + (1 - \partial)C_2, 0 < \partial < 1 \\ C'_2 = (1 - \partial)C_1 + \partial C_2, 0 < \partial < 1 \end{cases} \quad (3)$$

In order to improve the global search ability of population a, we cross the chromosomes with high similarity and reduce the probability of low similarity chromosome crossing. The specific implementation methods are referred to the literature [9].

When there is no effective gene in the population of genetic algorithm, the algorithm will tend to be local and precocious. In the model of web service composition, if the excellent candidate services in a candidate service set are missing, the algorithm will not be able to find the global optimal solution. Therefore, we introduce a big mutation operator in population B to increase the diversity of services.

For mutation operator, we define it as follows.

$$P = \begin{cases} P_{max} - \frac{(P_{max}-P_{min})(f-f_{avg})}{f_{max}-f_{avg}}, & f \geq f_{avg} \\ P_{max} & , f < f_{avg} \end{cases} \quad (4)$$

P_{max}, P_{min} represents the maximum and minimum probability of variation, respectively, and f_{avg} represents the average fitness.

When calculating the fitness value, we first normalize the QoS through expressions (1) or (2). At the same time, the service quality of each service is scored through the scoring function, and the scoring value of inferior service is artificially reduced through equation (7), which reduces the possibility of the service being selected in the subsequent roulette algorithm, and is conducive to the retention of excellent service. The random number α of $[0,0.3]$ is introduced to guarantee the randomness of the process, and the penalty value is modified.

The variables $(p_i^c, p_i^t, p_i^a, p_i^r)$ represent the cost, response time, availability and reputation of w_i . we_k represents the weight of the k th QoS attribute. The score function profit and fitness function fitness are as follows (5), (6).

$$Profit_i = \sum_k^4 we_k p_i^k \quad (5)$$

$$Fitness(ws_i) = Profit_i - D(i) \quad (6)$$

$$D(i) = \frac{P_{max}^a - P_i^a + P_{max}^r - P_i^r + P_i^t + P_i^c}{N} \alpha \quad (7)$$

$D(i)$ is the penalty value of the function for the i th service on a abstract service. With the increase of population iterations, the penalty value changes from large to small, which is conducive to reducing the probability of poor service inheritance in the early stage of evolution, and reducing the loss of excellent genes caused by too high penalty value in the later stage.

4. Simulation Experiment and Analysis

4.1 Simulation Environment

In order to test the performance and effectiveness of the improved dual population genetic algorithm, the simulation experiment of web service composition is implemented with Java language in the environment of CPU: 1.8GHz, 8.0GB RAM, windows10 operating system. Suppose there are eight abstract services in the web service composition problem, and each abstract service has 20-100 candidate servers, and the connection mode is sequential. By adjusting the number of candidate servers to observe the running time of the algorithm and the score of the selected service, the performance of the algorithm is judged.

4.2 Experiment Design

Three datasets are automatically generated from java random numbers, each of which contains 160, 480, 800 specific services, and are unevenly distributed in eight sequentially connected abstract task candidate sets. The performance and convergence of different algorithms are compared by changing the iteration times. The performance of the improved Multi-population genetic algorithm(MMGA), the general elite Multi-population genetic algorithm(MGA) and the general genetic algorithm(GA) are compared by profit. To be fair, each set of data is scored 200 times after the algorithm is executed. Table 2 compares the optimal services selected by the three algorithms after 20, 100 iterations of initial population. Each data in the table is an optimal solution selected

under the dataset. In this set of solutions, each specific service's QoS value is added to get the overall QoS value, and these QoS values are arranged in Cost/Availability/Time/Reputation order.

Table 2: Experimental results on datasets

	MMGA	MGA	SGA
Dataset1	223/1655/127/2032	223/1655/127/2032	223/1655/127/2032
Dataset2	167/1987/47/2509	149/1940/38/2319	138/1873/32/2418
Dataset3	124/2279/39/2681	125/2072/70/2768	118/1770/69/2759

From the above experimental results, it can be seen that when there are fewer services in the candidate server, the three algorithms can find the optimal solution, but when the number of services is increasing, the number of population is small, resulting in the lack of effective genes, which leads to the limitations of the solutions of the ordinary Multi-population genetic algorithm (MGA) and the ordinary genetic algorithm (SGA), while the improved Multi-population genetic algorithm (MMGA) obtains the global optimal solution.

In addition, we test the number of iterations needed by the three algorithms to find the optimal solution under the condition that the population number is 50 and the number of services is 800. Each group of data is the average of 100 times of algorithm execution. From figure 3, we can clearly draw a conclusion that when the number of iterations is less than 30, the score of service composition found by the MMGA is significantly higher than that found by the MGA and ordinary genetic algorithm (SGA), which proves the accuracy of the algorithm.

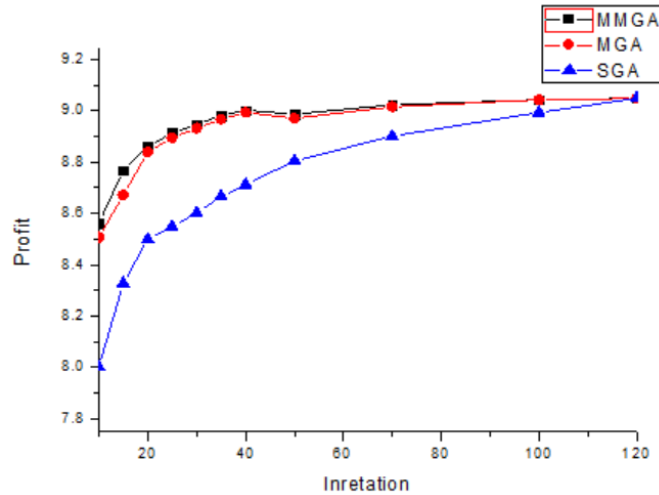


Figure 3: Comparison of scores between different algorithms

5. Conclusion

Based on the improved Multi-genetic algorithm in literature 9, a new framework of genetic algorithm is proposed to solve the problem of web service combination. By setting different genetic operators for two different populations, the algorithm shortens the evolutionary operation time, increases the diversity of individuals, and alleviates the contradiction between group diversity and population convergence. The early maturity phenomenon in the process of population evolution is avoided effectively. Compared with the common two population genetic algorithm, MMGA algorithm increases the possibility of excellent service under the premise of ensuring excellent local search. The negative effects of effective gene defects were reduced. Compared with the common dual

population genetic algorithm and legacy algorithm, a relatively efficient solution can be found under the same conditions, which can meet the user's QoS requirements to the maximum extent. However, the next step also needs to improve the adaptive performance of the algorithm and consider the impact of the execution time.

References

- [1] Deng S, Zhaohui WU. A survey of Web service composition methods [J]. Sciencepaper Online, 2008.
- [2] Chakraborty D, Perich F, Joshi A, et al. A Reactive Service Composition Architecture for Pervasive Computing Environments [J]. Ifip Advances in Information & Communication Technology, 2003, 106: 53-60.
- [3] Chifu V R, Pop C B, Salomie I, et al. Optimizing the Semantic Web Service Composition Process Using Cuckoo Search [C]// Intelligent Distributed Computing V- International Symposium on Intelligent Distributed Computing-idx. DBLP, 2011.
- [4] Liu B, Zhang R J. Web services composition method based on QoS by multiple objective optimization [J]. Computer Engineering and Design, 2012, 33 (3): 885-889. Ming C, Wang Z W. An Approach for Web Services Composition Based on QoS and Discrete Particle Swarm Optimization [C]// Eighth Acis International Conference on Software Engineering. IEEE Computer Society, 2007.
- [5] Zhang L J, Bing L, Tian C, et al. On demand web services [J]. IEEE, 2003, 4:4057 - 4064.
- [6] Ko J M, Chang O K, Kwon I H. Quality-of-service oriented web service composition algorithm and planning architecture [J]. Journal of Systems & Software, 2008, 81(11): 2079-2090.
- [7] Zhang C W, Su S, Chen J L. Genetic Algorithm on Web Services Selection Supporting QoS [J]. Chinese Journal of Computers, 2006, 29(7): 1029-1037.
- [8] Yu T, Lin K J. Service selection algorithms for Web services with end-to-end QoS constraints [J]. IEEE, 2005.
- [9] Liu Z M, Zhou J L, Li C, et al. A Novel Genetic Mutation Operator for Maintaining Diversity [J]. Mini-micro Systems, 2003, 24(5): 902-904.