Scientific Research Publishing

Journal of Computer and Communications, 2021, 9, 109-119
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Composition of Web Services of Multi-Population Adaptive Genetic Algorithm Based on Cosine Improvement

Siyuan Meng, Chuancheng Zhang

School of Computer Science and Technology, Shandong University of Technology, Zibo, China
Email: mengsiyuancm@gmail.com, zhangccxt999@gmail.com

How to cite this paper: Meng, S.Y. and Zhang, C.C. (2021) Composition of Web Services of Multi-Population Adaptive Genetic Algorithm Based on Cosine Improvement. *Journal of Computer and Communications*, **9**, 109-119.
https://doi.org/10.4236/jcc.2021.96006

Received: May 17, 2021
Accepted: June 18, 2021
Published: June 21, 2021

Abstract

Web quality of service (QoS) awareness requires not only the selection of specific services to complete specific tasks, but also the comprehensive quality of service of the whole web service composition. How to select the web service composition with the highest comprehensive QoS is a NP hard problem. In this paper, an improved multi population genetic algorithm is proposed. Cosine adaptive operator is added to the algorithm to avoid premature algorithm caused by improper genetic operator and the disadvantage of destroying excellent individuals in later period. Experimental results show that compared with the common genetic algorithm and multi population genetic algorithm, this algorithm has the advantages of shorter time consumption and higher accuracy, and effectively avoids the loss of effective genes in the population.

Keywords

Web Service Composition, Multi-Population Genetic Algorithm, QoS, Cosine Improved Adaptive Genetic Operator

1. Introduction

1.1. Research Significance and Background

With the increasing number of Web services in recent years, a large number of Web services were published on the network. Literature [1] and [2] pointed out that a single service has been unable to meet the needs of users, and more and more users need to combine web services to meet more needs. With the continuous development of network technology, users are not only satisfied with the realization of functional requirements, but also required to achieve good

non-functional requirements such as price, event, throughput and so on.

Web service composition is a process of selecting specific services from several abstract tasks and combining them into large granular services. According to the literature [3], from the perspective of business process, web service composition essentially refers to the connection of various tasks required by users in different ways. For a web service combination, each abstract task is completed by a specific service in the corresponding candidate service set. They have the same functions, but users can filter the services through a series of non-functional attributes: that is, the QoS attribute is used to constrain the service. However, there may be constraints among non-functional attributes. For example, when the price of service portfolio decreases, availability may also decrease. However, we need multiple indicators to judge the quality of services portfolio, and the problem will be transformed into multi-objective optimization. Therefore, it is a difficult problem to find a global QoS optimal solution under the premise of satisfying local constraints and completing the task flow.

1.2. Current Research Status

With the continuous development of intelligent optimization algorithms in recent years, more and more swarm intelligence optimization algorithms are studied and applied in Web service composition. For example, the cuckoo search based web service composition method proposed in reference [3], and the flying ant colony algorithm model proposed in reference [4] based on the improved ant colony algorithm to solve the multi-objective optimization and service selection problems in QoS, but at the same time, these methods can not avoid falling into the local optimal situation. Reference [5] proposed a new particle swarm optimization algorithm, bee colony optimization algorithm, to solve the multi-objective web service composition problem. Reference [6] used the improved NSGA-II algorithm to establish a multi-objective optimization model, and modified the population quality by crowding strategy. Combined with the above literature, I combine the crowding strategy with the elite strategy, and apply it to the algorithm proposed in this paper to ensure the quality of the selected individuals in the population. Combined with the idea of multimodality and niche in references [7] and [8], a new adaptive genetic algorithm MCGA is proposed to solve the problem of web service composition. The simulation results show that the algorithm has high superiority and feasibility, and can meet the actual needs.

2. Background Knowledge

2.1. QoS' Web Service Mix

2.1.1. Definition of QoS

Definition 1 Service quality (Quality of Service) refers to the set of requirements for network services in the process of network traffic transmission.

Among them, business flow refers to the grouping flow related to specific QoS, from source to purpose. Therefore, QoS is considered to be a set of

non-functional requirements for measurable Web services, which generally include response time (Time), price (Cost), availability (Availability), and reputation (Reputation).

Definition 2 in general, we can represent the problem of Web service combination by a quaternion (S, C_{in}, C_{out}, Q), S representing the finite set of services, C_{in} representing the input finite set of services, C_{out} representing the finite set of outputs, Q representing the finite set of service quality QoS.

2.1.2. QoS Related Calculations

The connection mode of each abstract service can be divided into parallel, sequential, and selected ways in the process of Web service combination oriented to business flow. For example, the $(w_1, w_2, \cdots, w_n)$ is connected in a sequential manner, while the $(w_1 \| w_2 \| \cdots \| w_n)$ is executed in parallel from w_1 to w_n. According to the above definition, we give the following Table 1, calculate the Web service QoS formula of different connection modes. Table 1 is shown below.

During the calculation of QoS, due to the different units of measurement of response time, availability and other attributes, the range of values is different, which will lead to the excessive proportion of some attributes and affect the overall results. Hence, we here normalize the QoS by formulas (1) and (2) and convert them to values of 0 to 1. For negative attributes such as response time and price, we normalize the attributes by (1). For positive attributes such as reputation, availability, throughput and so on, we normalize them by formula (2). For positive attributes such as reputation, availability and throughput.

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

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

2.2. Genetic Algorithms

2.2.1. Background and Basic Thought 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

Table 1. QoS formulas for composite services.

Link mode	Cost	Time	AVA	Rep
Sequence	$C=\sum_{i=1}^{n} C_i$	$T=\sum_{i=1}^{n} T_i$	$A=\prod_{i=1}^{n} A_i$	$R=\sum_{i=1}^{n} \frac{R_i}{n}$
Parallel	$C=\sum_{i=1}^{n} C_i$	$T=\max(T_1, T_2, \cdots, T_n)$	$A=\prod_{i=1}^{n} A_i$	$R=\sum_{i=1}^{n} \frac{R_i}{n}$
Selection	$C=\sum_{i=1}^{n} a_i C_i$	$T=\sum_{i=1}^{n} a_i T_i$	$A=\prod_{i=1}^{n} a_i A_i$	$R=\sum_{i=1}^{n} a_i R_i$

1970s. In the service computing problem, the service subscript is encoded and combined to represent the solution of a service composition on the chromosome. At the same time, the genetic operator in genetics is introduced to perform crossover and mutation operations, 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. How to choose the genetic operator greatly affects the superiority of the algorithm.

2.2.2. General Process of Genetic Algorithm

Obviously, fixed crossover and genetic operators can not meet our needs, so Srinvas *et al.* proposed adaptive genetic algorithm [9] the reference (Adaptive GA, short for AGA). The crossover rate and variation rate are linearly adjusted between the average fitness and the maximum fitness of the population, as shown in (3), (4) below.

$$P_C = \begin{cases} \dfrac{k_1\left(f_{\max} - f'\right)}{f_{\max} - f_{\text{avg}}} & f' \geq f_{\text{avg}} \\ k_2 & f' < f_{\text{avg}} \end{cases} \tag{3}$$

$$P_m = \begin{cases} \dfrac{k_3\left(f_{\max} - f\right)}{f_{\max} - f_{\text{avg}}} & f \geq f_{\text{avg}} \\ k_4 & f < f_{\text{avg}} \end{cases} \tag{4}$$

$f_{\max}$ represents the maximum fitness of the population, f_{avg} represents the average fitness of the population, f' represents the larger fitness in the cross operation, and f represents the fitness of the individual.

From the formulas (3) and (4), it is not difficult to see that the optimal individuals in the population are difficult to change at the beginning of the algorithm. Therefore, the simple linear adaptive operator is easy to make the algorithm converge to the excellent individuals at a certain stage. It is obvious that the optimal individual in this stage is not necessarily the global optimal, so we can see that the above linear adaptive operator still has the possibility to converge the algorithm locally. In this paper, a cosine adaptive genetic operator can be used to solve this family problem effectively.

Through the above analysis, we can conclude that the traditional genetic algorithm has the problem of premature convergence, and the population is easy to lose the diversity of genes in the early stage and fall into the problem of local search for the optimal solution. In the early stage, the individual fitness in the population is low, the smaller genetic operator is not easy to produce the excellent individual, but in the later stage of the algorithm, the individual adaptation value in the population is higher. It will destroy the good individuals in the population. Therefore, we need a genetic operator that can change with the algorithm.

In addition, when the amount of data is too large or the number of individuals

in the population is not enough, the probability of the emergence of excellent genes will be reduced, so that the optimal solution can not be obtained. For example, if an abstract service has too many specific services, fewer individuals and low variation rates reduce the possibility of optimal gene emergence. Based on the above problems, this paper proposes a Web service combination method based on multiple group genetic algorithms to solve the problem, which can make the algorithm jump out of the local extremum and create a new search plane, which effectively avoids the precocity of the algorithm.

3. Algorithm Design and Implementation

3.1. Chromosome Representation

In the algorithm proposed in this paper, a chromosome represents a set of service combinations, abstract services represent a gene locus on the chromosome, and the number of abstract services is the number of loci. Each specific service can be represented as a gene. We put similar specific services into a service set to generate a candidate service set. Each locus will correspond to a specific set of candidate services to achieve a task in the business process by selecting a specific service in the set of candidate services. Select a service on each locus to form a chromosome, which is represented as a set of service combinations.

In terms of chromosome structure and gene coding, an integer encoding method is similar to that used in the reference [8], A chromosome structure contains N loci, each locus holds Type, connection of abstract services subscript Index, of specific services in the service candidate set and the attributes that measure the quality of service calculated by fitness function in formula (3) profit. On each gene of the locus, having several QoS properties representing the current service, Time, Cost, Reputation, Availability. The chromosome structure is shown in **Figure 1**.

3.2. Cosine Adaptive Operator

The genetic operator design of genetic algorithm has a very important influence

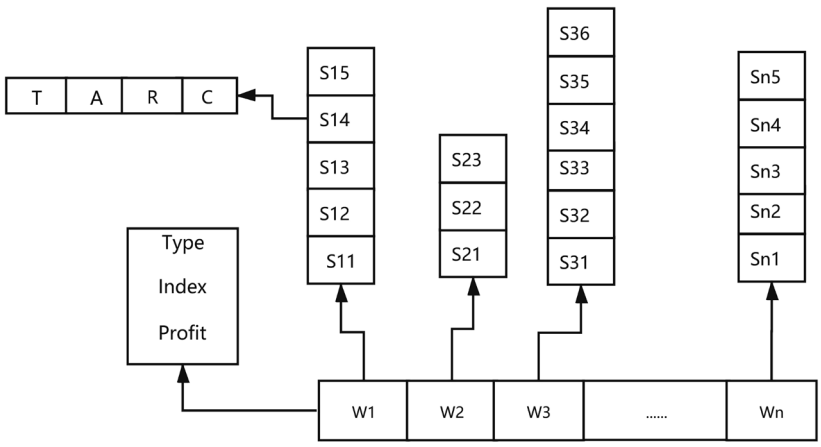

Figure 1. Chromosome structure.

on the performance of the algorithm. In order to solve the phenomenon of premature and slow convergence of the algorithm, a cosine improved adaptive genetic algorithm CAGA proposed in reference [10]. The genetic operators constructed are as follows (5), (6):

$$P_c = \begin{cases} \dfrac{p_{c\max} + p_{c\min}}{2} + \dfrac{p_{c\max} - p_{c\min}}{2} \cos\left(\dfrac{f' - f_{\text{avg}}}{f_{\max} - f_{\text{avg}}}\pi\right) & f' \geq f_{\text{avg}} \\ p_{c\max} & f' < f_{\text{avg}} \end{cases} \tag{5}$$

$$P_m = \begin{cases} \dfrac{p_{m\max} + p_{m\min}}{2} + \dfrac{p_{m\max} - p_{m\min}}{2} \cos\left(\dfrac{f - f_{\text{avg}}}{f_{\max} - f_{\text{avg}}}\pi\right) & f \geq f_{\text{avg}} \\ p_{m\max} & f < f_{\text{avg}} \end{cases} \tag{6}$$

The expression $[\, p_{c\min}, p_{c\max}, p_{m\min}, p_{m\min} \,]$ represents the minimum crossover rate, the maximum crossover rate, the minimum mutation rate and the maximum mutation rate respectively. Compared with AGA, CAGA, it reduces the possibility of precocity and increases population diversity. Meanwhile CAGA there is no superiority in dealing with the individual between the average fitness value and AGA maximum fitness value. Because of the characteristics of cosine function near (0, π) CAGA algorithm can increase the mutation rate and crossover rate of individuals in the range of $\left[f_{\text{avg}}, \dfrac{f_{\text{avg}} + f_{\max}}{2}\right]$, accelerate the convergence of the algorithm, and reduces the variation rate and crossover rate of individuals in the range of $\left[\dfrac{f_{\text{avg}} + f_{\max}}{2}, f_{\max}\right]$, maintain good individuals as well.

3.3. Improved Double Population Genetic Algorithm Implementation

In the process of convergence of genetic algorithm, with the iteration, the individuals in the population will become more and more single, and the population diversity will decrease, so there will be a situation of falling into the local optimal solution. Most of the reasons for this are the lack of excellent genes in the population, which leads to the algorithm only finding 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-group structure and elite multi-population structure to obtain the optimal solution through the convergence of elite population.

When the population is initialized separately, multiple populations execute the MCGA algorithm separately, and each generation of excellent individuals is retained into the elite population. When each population MCGA executed, it is executed again in the elite population MCGA, and the optimal individual is obtained by judging the conditions. **Algorithm 1** illustrates the steps of MCGA.

In genetic operation, we use single point crossover to minimize the possibility of reducing individual fitness. The specific operations are as follows: C_1, C_2 represents the chromosomes operated for the parent generation, C_1', C_2' represents

Algorithm 1. Multi-population cosine genetic algorithm.

1 Intilization: initialize each m populations $P_1 \sim P_m$ randomly
2 **While** ($t \le m$) do
3 **for** ($q = 0$ to max generation n) **do**
4 Evaluate fitness of each chromosomes
5 Pick chromosomes $X^i \sim X^i$ by roulette wheel selection algorithm where ($X^i \ne X^i$)
6 Corssover operator to P_t^q
7 Mutation operator to P_t^q
8 Add best solution X^e to elite population P_e and q++
9 **end for** and t++
10 **end while**
11 **for** ($q = 0$ to max generation n) **do**
12 Evaluate fitness of each chromosomes
13 Selection operator to P_e
14 Corssover operator to P_e
15 Mutation operator to P_e
16 q++
17 **end for**

the chromosomes after the cross operation, and ∂ is the part of the crossover. When we choose individuals to cross, we apply crowding strategy to individual selection strategy. Individuals with higher crowding degree were selected as the parents of the next generation population.

$$\begin{cases} C_1' = \partial C_1 + (1-\partial) C_2, & 0 < \partial < 1 \\ C_2' = (1-\partial) C_1 + \partial C_2, & 0 < \partial < 1 \end{cases} \tag{7}$$

when the population of genetic algorithm lacks effective genes, the algorithm tends to be local and precocious. If the excellent candidate service on a candidate service set is missing in the model of Web service combination, the algorithm must not find the global optimal solution. Therefore, we use the two cosines adaptive operator mentioned above and design the crossover operator and genetic operator according to formula (3) and formula (4) respectively.

When calculating the fitness value, we first normalize the QoS by expression (1) and expression (2) respectively. At the same time, the service quality of each service is scored by the scoring function (5), and the fitness value is calculated by formula (6). At the same time. The higher the score, the better the solution of the service composition.

$$Profit_i = \sum_k^4 we_k p_i^k \tag{8}$$

$$\text{Fitness}(ws_i) = Profit_i - D(i) \tag{9}$$

In order to optimize the selection of individuals by roulette algorithm, we use penalty function (7) to reduce the fitness value of inferior services to reduce the possibility of the individual being selected in the subsequent roulette algorithm. In the later stage of the algorithm, it is helpful to retain excellent service group merging and eliminate inferior service combination in the early stage, which speeds up the convergence of the algorithm. In order to ensure the randomness of the process, the ∂ random number of [0.0.2] is introduced into expression (7),

and the penalty value is corrected. Variables ($p_i^c, p_i^t, p_i^a, p_i^r$) represents price, response time, availability, reputation of w_i, respectively. we_k represents the weight of the k QoS attribute.

$$D(i) = \frac{P_{\max}^a - P_i^a + P_{\max}^r - P_i^r + P_i^t + P_i^c}{N} \partial \tag{10}$$

4. Simulation Experiment and Analysis

4.1. Simulation Environment

Aiming to test the performance and effectiveness of the improved two-population genetic algorithm, At CUP: 1.8 GHz, 8.0 GB RAM, Windows10 operating system, under the 10 Mbp/s environment, the simulation experiment of Web service combination is realized by using Java language. Assuming there are eight abstract services in the Web service portfolio problem, each abstract service has 20 to 100 candidate servers, the connection mode adopts sequential connection. By adjusting the number of candidate servers to observe the running time of the algorithm and select the service score to judge the performance of the algorithm.

4.2. Experimental Design

Automatically generate three data sets by java random numbers, each data set includes 160, 480, 640, 800 specific services, unevenly distributed in eight sequential connected abstract task candidates, by changing the iteration, Compare the performance and convergence between different algorithms. Comparison of Cosine Adaptive Multiple Group Genetic Algorithms (MCGA) by Scoring, the performance of common adaptive elite multi-group genetic algorithm (MAGA) and common multi-group genetic algorithm (MGA) is discussed. To be fair, the score of each group of data is the average time after the algorithm executes 200. Table 2 compares the three algorithms with a population size of 20 in the initial test, the optimal service selected after 100 iterations. Each item in the table is an optimal solution selected under the data set. In this set of solutions, The QoS value of the specific service corresponding to each abstract service is added together to obtain the overall QoS value of the service combination, Arrange them in Cost/Availability/Time/Reputation order.

We can see from the above experimental results 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 comprehensive QoS value of

Table 2. Experimental results on the dataset.

	MCGA	MAGA	MGA
Dataset1	220/1355/107/2432	220/1355/107/2432	220/1355/107/2432
Dataset2	162/2087/47/2509	144/1970/38/2319	135/1873/32/2418
Dataset3	124/2279/39/2681	125/2072/70/2768	118/1770/69/2759
Dataset4	120/2012/51/2732	123/2022/73/2732	119/1992/82/2698

the MCGA is obviously higher than that of the other two algorithms. since the number of experimental populations is small, the possibility of deletion of effective genes in the initial population increases, leading to MGA and MAGA precocious in the initial stage of the algorithm and beginning to converge to the local optimal solution. However, MCGA the fitness of individuals near the average value is effectively increased by cosine adaptive genetic operator, which solves the problem that the algorithm can not jump out of the search plane to obtain the global optimal solution when the average fitness and the maximum fitness are close.

Moreover, we test the number of iterations when the MCGA, MAGA, MGA converges on the Dataset2, Dataset3, Dataset4 under the condition that the population number is 50, and each set of data is averaged after the algorithm executes 100 times. Figures 2-4 show that when the number of iterations is small,

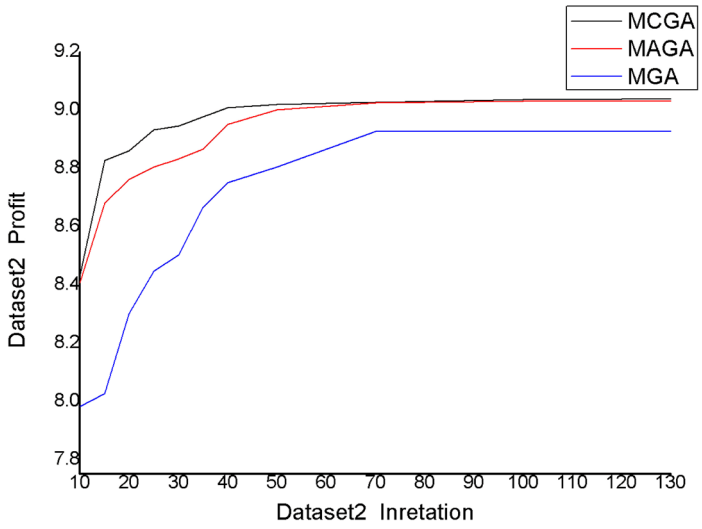

Figure 2. Data set 2 experimental results.

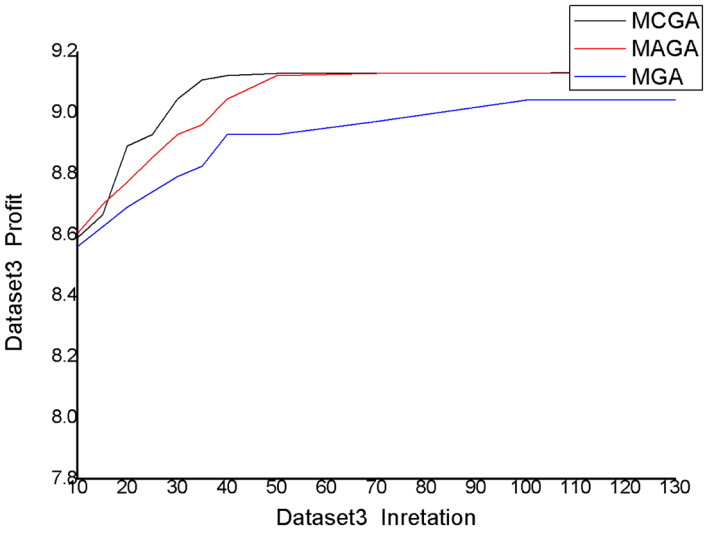

Figure 3. Data set 3 experimental results.

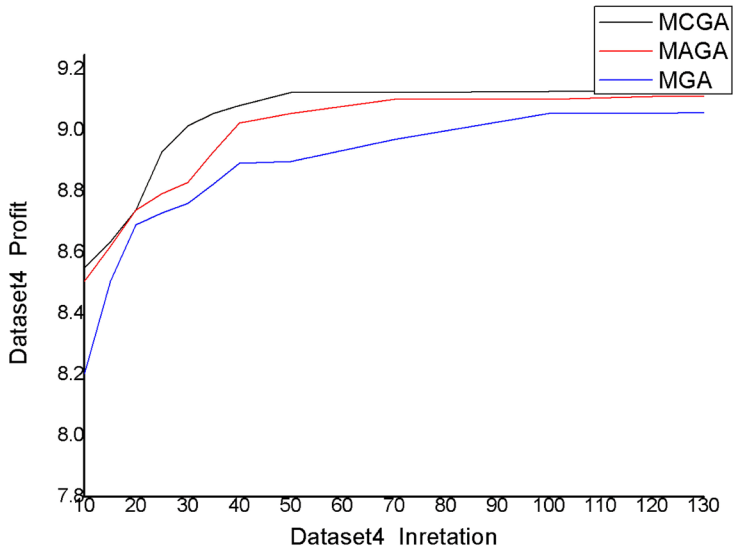

Figure 4. Data set 4 experimental results.

the service combination solution found by MCGA is obviously superior to MAGA and MGA. With the increase of iteration number, it is obviously easier to converge locally in dataset 1. Both MCGA and MAGA can find an optimal solution. On the other hand, in two data sets with more services, MAGA the solution near the average value is not processed carefully enough, the convergence speed of the algorithm is obviously slower than MCGA, and it is easier to fall into the local optimal solution. The average value of the solution found by the MAGA is obviously not as good as that of the algorithm MCGA, which proves the accuracy and efficiency of the MCGA.

5. Concluding Remarks

In this paper, a genetic algorithm based on improved adaptive genetic operator is proposed to solve the service composition problem. Elitism and crowding strategy are used to ensure the accuracy and convergence speed of the algorithm. Through the improved cosine genetic operator set for multiple different populations, the algorithm shortens the operation time of evolution, increases the diversity of individuals, effectively improves the convergence speed of the algorithm, and avoids the premature situation at the same time. Compared with the common two-population genetic algorithm MGA, and the improved adaptive genetic algorithm MAGA, MMGA the mutation rate and crossover rate of individuals near $\left[f_{\text{avg}}, \frac{f_{\text{avg}}+f_{\max}}{2}\right]$ are increased on the premise of excellent local search. The mutation rate and crossover rate of individuals near $\left[\frac{f_{\text{avg}}+f_{\max}}{2}, f_{\max}\right]$ are reduced. However, there are still some problems in the adaptive operator of cosine improvement. When the gap between $f_{\text{avg}}, f_{\max}$ is too large, the adaptive adjustment curve of MCGA tends to be linear, so it needs further correction.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Barreto, C., Bullard, V., Erl, T., *et al.* (2007) Web Services Business Process Execution Language Version 2.0. Medicina.

[2] Konak, A., Coit, D.W. and Smith, A.E. (2006) Multi-Objective Optimization Using Genetic Algorithms: A Tutorial. *Reliability Engineering & System Safety*, **91**, 992-1007. https://doi.org/10.1016/j.ress.2005.11.018

[3] Deng, S. and Zhaohui, W.U. (2008) A Survey of Web Service Composition Methods. *Sciencepaper Online*, **3**, 23-27.

[4] Dahan, F., Hindi, K.E., Ghoneim, A., *et al.* (2021) An Enhanced Ant Colony Optimization Based Algorithm to Solve QoS-Aware Web Service Composition. *IEEE Access*, **9**, 34098-34111. https://doi.org/10.1109/ACCESS.2021.3061738

[5] Seghir, F. (2021) FDMOABC: Fuzzy Discrete Multi-Objective Artificial Bee Colony Approach for Solving the Non-Deterministic QoS-Driven Web Service Composition Problem. *Expert Systems with Applications*, **167**, Article ID: 114413. https://doi.org/10.1016/j.eswa.2020.114413

[6] Sadeghiram, S., Ma, H. and Chen, G. (2020) QoS-Constrained Multi-Objective Distributed Data-Intensive Web Service Composition—NSGA-II with Repair Method. *Proceedings of the* 2020 *Genetic and Evolutionary Computation Conference Companion*, Cancun, 8-12 July 2020, 105-106. https://doi.org/10.1145/3377929.3389977

[7] Lin, D., He, L., Feng, X., *et al.* (2018) Niching Pareto Ant Colony Optimization Algorithm for Bi-Objective Pathfinding Problem. *IEEE Access*, **6**, 21184-21194. https://doi.org/10.1109/ACCESS.2018.2822824

[8] Yang, Q., Chen, W.-N., Yu, Z.T., *et al.* (2017) Adaptive Multimodal Continuous Ant Colony Optimization. *IEEE Transactions on Evolutionary Computation*, **21**, 191-205. https://doi.org/10.1109/TEVC.2016.2591064

[9] Srinivas, M. and Patnaik, L.M. (1994) Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, **24**, 656-667. https://doi.org/10.1109/21.286385

[10] Kuang, H. (2006) Improving Crossover and Mutation for Adaptive Genetic Algorithm. *Computer Engineering and Applications*, **12**, 93-96.