Novel Algorithm for Composite Service Choosing

Zhiwu Wang[1]
Department of Computer Science and Engineering, Henan Institute of Engineering, Zhengzhou, P.R.China
Email: wangzhiwu1974@gmail.com

Yanhui Zhang[2*]
School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing, P. R.China
Email: huihui2001922@163.com

***Abstract*—Among a variety of composite services meeting functional demands, how to make optimization choice is a difficult problem faced by users whose QoS demand is a multidimensional objective, and QoS features are not interdependent. Therefore, the service choice based on QoS aggregation is a typical multi-objective optimization problem. By analyzing the existing optimization choice algorithm for composite service based on QoS aggregation, this paper has proposed an optimization choice algorithm for composite service based on cooperative evolutionary genetic algorithm; on the basis of defining service QoS attribute feature vector, it analyzes the strategy and mechanism influencing the efficiency and solution space of this algorithm and verifies the effectiveness and feasibility of this algorithm by comparison with the experiment on traditional single-species genetic algorithm.**

***Index Terms*—Composite service, The QoS aggregation, Service choice, Co-evolution genetic algorithm**

I. Introduction

Making evaluation on the non-function property QoS of composite service is the basis for users to make optimization choice for composite service. Users' QoS demand is a multidimensional objective and QoS features are not interdependent. Therefore, the service choice based on QoS aggregation is a typical multi-objective optimization problem. The current algorithms for solving the optimization problem of composite service based on QoS aggregation are divided into two categories: The first category adopts traditional multipurpose optimal methods such as Linear Programming[1], Analytic Hierarchy Process[2], TOPSIS[3], Markov chain[4] and so on to realize research for optimum solution. When using this algorithm, we must obtain the optimum solution of the problem through exhaustive solution space, and the complexity of time computation will get great by exponent along with the increase of the number of candidate physical service in the efficient solutions of composite service, to make it difficult to meet the real-time demands. The other category adopts the intelligent optimization algorithms such as Traditional Genetic Algorithm[5][6], Swarm Optimization Algorithm[7], Simulated Annealing Algorithm[8] and Ant Colony Algorithm[9][10] to solve the locally optimum solution of service composition, which realizes the multi-objective local optimization by establishing the Pareto optimal solution set of the problem and meets the needs of the optimization problem of physical composite service, but the problems such as convergence and solutions distribution must be solved for the algorithm.

Reference[11] has given the selection algorithm of single path service combination, using GA of one-dimensional chromosome encoded mode; Reference[12] has given the chromosome encoded mode, fitness function and trigger algorithm for rescheduling Web combination service path in dynamic environment, and carried out simulation verification, also using one-dimensional chromosome encoded mode; Reference[13] has given a genetic algorithm based on relational matrix encoded mode to solve Web service selection based on Qos. These researches just designed the encoded mode of GA; besides that, crossover strategy, mutation strategy and other parts of GA should also be designed based on the encoded mode to realize practical application in service selection. Based on the analysis for the above algorithms and in connection with the features of combination service based on QoS aggregation, this paper proposes an optimization algorithm for combination service based on cooperative evolutionary genetic algorithm to realize the optimization choice of the efficient solutions of physical combination service, regarding the feature vector of QoS aggregation as objective function. The algorithm designed by making a comprehensive research and design of encoding strategy, crossover and mutation strategy, fitness function structure, collaboration among subpopulations, external file update, structure of next-generation subpopulations and other aspects of GA can achieve the optimum selection of combination service under the mode of complex control flow and get better optimum Pareto solutions more quickly, compared to the traditional genetic algorithm.

Manuscript received Jan 11, 2011; revised April 29, 2011; accepted May 26, 2011.

*Corresponding author.

doi:10.4304/jsw.7.1.161-168

In the second part, this paper has given a definition to the feature vector of service QoS; in the third part, this paper has proposed an optimization choice algorithm for composite service based on Improved Multi-Objective Cooperative Evolutionary Genetic Algorithm (IMOCEGA) and analyzed the strategy and mechanism influencing the efficiency and solution space of the algorithm; the fourth part has verified the effectiveness and feasibility of this algorithm through the analysis on the comparative trial of the algorithm mentioned in this paper and the traditional single-species genetic algorithm; the fifth part is a conclusion of the paper.

II. DEFINITION AND PRETREATMENT OF QOS FEATURE VECTOR

This paper has proposed to regard feature vector of extensible multidimensional QoS as the multi-objectives decision variable of optimization algorithm of composite service. Among the formula $V_{QoS}=\{VP,VST,VA,VSR,VR,VS\}$. The six decision variables are price V_P, service-time V_{ST}, availability V_A, success ratio V_{SR}, reputation V_R and security V_S respectively.

1) Price: Price means the expenses shall be paid when the task is solved successfully by the service.

2) Service-time: Service-time means the total time used by the system from that the task is submitted to the service to that the service completes the execution and returns the results to users.

The service-time of the service is composed of the three parts, respectively service-communicate-time, service-execute-time and component-ready-time, as shown in formula 1[2], of which $T_{service\text{-}communicate}$ is the sum of data transmission time and network delay time; $T_{service\text{-}execute}$ is calculated and obtained from the statistical value of the execution time of history services; $T_{component\text{-}ready}$ refers to the overhead time of related components startup during the service execution, such as the setup time of task dispatcher.

$$V_{ST} = T_{service-communicate} + T_{component-ready} + T_{service-execute}$$

$$T_{service-communicate} = \frac{Size_{datainput} + Size_{dataoutput}}{Bandwidth} + T_{latency} \quad (1)$$

3) Availability: Availability means the current callable probability of physical services.

The availability of service can be calculated by formula 2[4], of which, $T_{\text{time-to-failure}}$ represents the mean time between failures of service operation, while $T_{time\text{-}to\text{-}recover}$ represents the mean-time-to-repair, and both can be obtained through the execution log statistics of the system.

$$V_A = \frac{T_{time-to-failure}}{T_{time-to-failure} + T_{time-to-recover}} \quad (2)$$

4) Success-ratio: Success-ratio means the probability of completing a task within a period of time after the service is established.

The success ratio of service can be calculated from the times number ($N_{Success}$) of the successful execution of service at unit time and the number of times the service is called ($N_{Success}$+$N_{failure}$)[2], as shown in formula 3.

$$V_{SR} = \frac{N_{Success}}{N_{success} + N_{failure}} \quad (3)$$

5) Reputation: Reputation means the users' trust and satisfaction for service and its values come from the evaluation of users about the practical result of called service.

The service reputation adopts the evaluation system of fuzzy marking[3] with the values of fuzzy literals *VH (Very High), H (High), M (Middle), L (Low)* and *VL (Very Low)*, mapping the fuzzy marks as the corresponding triangular fuzzy numbers by the quantitative way to eliminate the subjective error of users when they evaluate the service reputation. The gravity centers of triangular fuzzy numbers are used as the precise values of reputation, as shown in formula 4, of which, $\lambda(x)$ represents the membership function of R and S represents the integral interval.

$$V_{reputation_i} = \frac{\int_S x\lambda(x)dx}{\int_S \lambda(x)dx} = \frac{\int_{R^L}^{R^M} x\lambda(x)dx + \int_{R^M}^{R^U} x\lambda(x)dx}{\int_{R^L}^{R^M} \lambda(x)dx + \int_{R^M}^{R^U} \lambda(x)dx} \quad (4)$$

The total reputation of service shall be solved with mean value method, shown in formula 5, of which, V_R represents the precise value of service reputation evaluation from the user calculated by formula 4 and n represents the total number of users who has called the service.

$$V_{\mathrm{R}} = \frac{\sum_{i=0}^{n} V_{reputation_i}}{n} \quad (5)$$

6) Security: Security means the security class of the service, including the security of data resource access control policies adopted by the service and the security of data transmission encryption policies adopted by the service.

TABLE I.
SERVICE SECURITY LEVEL

Security level of data access control policies		Security level of data transmission encryption policies	
Data access control granularity	Security level	Key length of data encryption	Security level
Access control based on role	1	56	1
Access control based on table	2	128	2
Access control based on lines	3	168	3
Access control based on data items	4	192	4

This paper has divided the security values of access control policies into 4 levels according to the granularity of data resource access control policies, and in the same way, this paper has divided the values of data transmission encryption policies into 4 levels according to the key length of data encryption, as shown in Table Ⅰ.

III. Optimization Choice Algorithm for Composite Service Based on QoS Aggregation

A. QoS solution under basic control flow model

Among the efficient solutions of composite service of a task, the calculation of QoS values of all QoS features is related not only to the QoS values of single-member service but also to the control flow model of itself. Generally speaking, composite service has seven basic control flow models including sequence, parallel, circulation, selection, discriminator, synchronization and multi-instance models, which are combined to realize the internal structure of any complicated composite service. Jaeger[14] has given the calculation formula of QoS aggregation for solving the service price, response time and security under partial models, but it lacks QoS aggregation analysis under multi-example model. To accomplish the calculation of the QoS value for efficient solution of composite services, this paper has extended the calculation formula and has successively given the QoS aggregation calculation formula under seven basic control flow models, as shown in Table Ⅱ. (*n* is the number of member services existing in the model)

TABLE II.
CALCULATION FOMULA OF QoS AGGREGATION UNDER BASIC CONTROL FLOW MODEL

QoS Control Flow	Price	Service-Time	Availability	Success-Ratio	Reputation	Security
Sequential	$\sum_{i=1}^{n} V_{\mathrm{P}}^{P}$	$\sum_{i=1}^{n} V_{S}^{P}$	$\prod_{i=1}^{n} V_{A}^{P}$	$\prod_{i=1}^{n} V_{S}^{P}$	$\min(V_{\mathrm{R}}^{P})$	$\min(V_{S}^{P})$
Parallel	$\sum_{i=1}^{n} V_{\mathrm{P}}^{P}$	$\max(V_{S}^{P})$	$\prod_{i=1}^{n} V_{A}^{P}$	$\prod_{i=1}^{n} V_{S}^{P}$	$\min(V_{\mathrm{R}}^{P})$	$\min(V_{S}^{P})$
Circulation	nV_{P}^{P}	nV_{S}^{P}	$(V_{A}^{P})^{n}$	$(V_{S}^{P})^{n}$	V_{R}^{P}	V_{S}^{P}
Choice	$\frac{1}{n}\sum_{i=1}^{n} V_{\mathrm{P}}^{P}$	$\frac{1}{n}\sum_{i=1}^{n} V_{S}^{P}$	$\frac{1}{n}\sum_{i=1}^{n} V_{A}^{P}$	$\frac{1}{n}\sum_{i=1}^{n} V_{S}^{P}$	$\frac{1}{n}\sum_{i=1}^{n} V_{\mathrm{R}}^{P}$	$\frac{1}{n}\sum_{i=1}^{n} V_{S}^{P}$
Discriminator	$\sum_{i=1}^{n} V_{\mathrm{P}}^{P}$	$\min(V_{S}^{P})$	$\prod_{i=1}^{n} V_{A}^{P}$	$\prod_{i=1}^{n} V_{S}^{P}$	$\min(V_{\mathrm{R}}^{P})$	$\min(V_{S}^{P})$
Synchronization	$\sum_{i=1}^{n} V_{\mathrm{P}}^{P}$	$\max(V_{S}^{P})$	$\prod_{i=1}^{n} V_{A}^{P}$	$\prod_{i=1}^{n} V_{S}^{P}$	$\min(V_{\mathrm{R}}^{P})$	$\min(V_{S}^{P})$
Multiple Instance	nV_{P}^{P}	V_{S}^{P}	$(V_{A}^{P})^{n}$	$(V_{S}^{P})^{n}$	V_{R}^{P}	V_{S}^{P}

The source and explanation of the QoS aggregation calculation formula under seven control flow modes are as follows:

1) Sequential mode: The sequential mode means that member services are carried out one by one. As all member services under sequential mode are to be carried out in sequence, the price and service response time are obtained through summation; the availability and success ratio are obtained through quadrature according to probability; the reputation and security are relatively independent among member services and are obtained by adopting minimums.

2) Parallel mode: The parallel mode means that member services are carried out in parallel simultaneously. Different from sequential mode, as member services under parallel mode are carried out simultaneously, its response time should be the maximum among all member services under this mode.

3) Circulation mode: The circulation mode means that some member service is carried out repeatedly n times. In fact, we can regard the circulation mode as a special sequential mode made of n identical member services, so the calculation of its QoS aggregation is similar to that of sequential mode.

4) Choice mode: The choice mode means that there are branches in the implementation of multiple member services and only the member services in the branch which meets conditions will be carried out, with the other branches abandoned. As only one branch will be selected to be carried out under choice mode, presuming there are n branches and each branch has equal probability to be selected, the probability of being selected of each branch is *1/n*; then QoS aggregation under choice mode can be obtained by taking this value into calculation.

5) Discriminator mode: As member services are carried out in parallel, the calculation of QoS aggregation under the discriminator mode is similar to that of parallel mode; their difference is that under discriminator mode follow-up member services will be carried out as long as a single member service is completed, so its response time should be the minimum among all member services under this mode.

6) Synchronization mode: It means that there is communication synchronization among multiple member services in the implementation process, also meaning that some member service will suspend when carried out to some procedure and wait for synchronization confirmation from another member service. As member services under synchronization mode are carried out in parallel, its QoS aggregation vector is in line with that of parallel mode. Under this mode, the actual overall response time may be larger than the maximum among the response time of all member service, which arises from that there may be waiting time for communication synchronization among member services and the time for communication synchronization is neglected because the waiting time for communication synchronization cannot be obtained in design and it is always shorter than service response time.

7) Multiple-instance mode: Under this mode, the instance number (*n*) of some member service conforming to functional requirements of the task is determined according to QoS demands of the task and

these multiple instances are allocated to multiple resources to be carried out in parallel. Under the multiple-instance mode, the overall response time should be the response time of single member service instance, plus transmission and allocation time of instances; as the allocation time of instances is far shorter than the implementation time in normal circumstances, the allocation time is neglected.

Through the calculation of QoS aggregation under the basic control flow model, we get the QoS aggregation vector of efficient solution of the service, which is: $V_{QoS}^{T}=\left\{V_{P}^{T},V_{ST}^{T},V_{A}^{T},V_{SR}^{T},V_{R}^{T},V_{S}^{T}\right\}$.

B. Optimization choice algorithm for composite service based on IMOCEGA

A typical optimization algorithm for composite service based on QoS aggregation is expressed as follows: Suppose if each efficient solution of composite service has n abstract services, then $ADSResult=\{ADS_1,ADS_2,...,ADS_n\}$ is one efficient solution of composite service, of which, ADS_j is an abstract service of *ADSResult*. Suppose if each abstract service has m physical services, which is $ADS_j=\{PDS_1,PDS_2,...,PDS_m\}$, then we can get the efficient solution set of physical composite service of *DT*: $PDSResultSet=\{PDSResult_1, PDSResult_2,..., PDSResult_i\}$, of which, $t=n^m$, means t efficient solutions of physical composite service can meet the functional requirements of task *DT*. To get the physical composite service with optimum QoS, we must calculate the QoS aggregation decision vector of the efficient solution of the tth physical composite service, and chose the efficient solution of physical composite service with optimum QoS aggregation decision vector, as shown in Fig.1. When n and m are small, this process can be obtained through exhaustion method, but when n and m get great rapidly, the time complexity for calculation will increase by exponent. Due to the real-time requirement for task solution, it is very important to rapidly find an efficient solution of local optimum physical composite service for the users' solution precision.

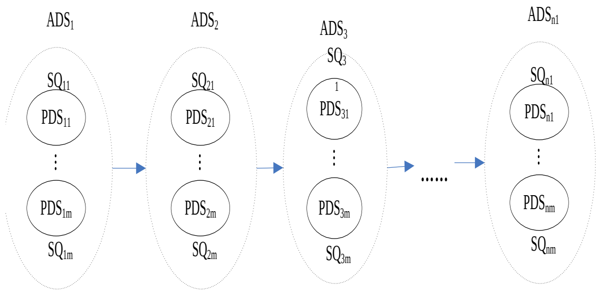

Figure1. Acquisition of the efficient solutions of optimum physical composite service based on QoS

For this reason, this paper has proposed an optimization choice algorithm for physical composite service based on Improved Multi-Objective Cooperative Evolutionary Genetic Algorithm (IMOCEGA) to realize the optimum solution of Pareto. By dividing the multi-objective optimization problem into many sub-problems of single-objective optimization and producing one subspecies for each sub-objective, IMOCEGA adopts the cooperative evolution of many subspecies and completes the interactivity of many subspecies through the integration between individuals in the subspecies and the representatives in other species, to eventually get the optimum Pareto solution of the problem.

$$\begin{cases}\min\ f_1((V_{\mathrm{P}})_{s1},(V_{\mathrm{P}})_{s2},...,(V_{\mathrm{P}})_{sn})\\ \min\ f_2((V_{ST})_{s1},(V_{ST})_{s2},...,(V_{ST})_{sn})\\ \max\ f_3((V_{A})_{s1},(V_{A})_{s2},...,(V_{A})_{sn})\\ \max\ f_4((V_{SR})_{s1},(V_{SR})_{s2},...,(V_{SR})_{sn})\\ \max\ f_5((V_{\mathrm{R}})_{s1},(V_{\mathrm{R}})_{s2},...,(V_{\mathrm{R}})_{sn})\\ \max\ f_6((V_{S})_{s1},(V_{S})_{s2},...,(V_{S})_{sn})\\ si\in ADS_i\end{cases} \quad (6)$$

A typical optimization choice problem of physical composite service can be expressed with formula 6.

Multi-objective optimization model includes six QoS decision objectives which correspond to the six QoS features of QoS aggregation decision vector, of which, to respectively are the calculation formulas for solving the QoS values of all QoS features in the efficient solutions of physical composite service, composed of the calculation formulas in Table Ⅱ. For consumptive QoS features such as service price and service response time, the optimum objectives of them are minimum values. For the efficient QoS features such as availability, success ratio, reputation and security, the optimum objectives of them are maximum values.

The basic process of the optimization choice algorithm for physical composite service based on improved multi-objective cooperative evolutionary genetic algorithm is as follows:

Step1: Initialization. Set the number of subspecies as N according to QoS decision objectives, randomly to create N initial subspecies with the scale of M; set the number of evolution generations as 1, set an external file of blank and set the maximum capacity of the external file as L, among which the external file is used to save the complete optimum Pareto solution obtained from the algorithm during the evolution;

Step2: Make orderly such genetic evolution operations as crossover operation and mutation operation for each subspecies, to obtain the filial generations of all subspecies; Choose different mutation probabilities according to the similarity of the individuals in the subspecies at the time of mutation;

Step 3: make cooperative operation for each individual of all the subspecies to create complete solutions and get the completely evolved individuals of the whole subspecies; Estimate the adaptive value of the complete individual and determine whether this individual needs an update or abandonment of external file, till all the individuals in all subspecies are completely treated;

Step 4: If the actual number of the complete solutions in the external file is larger than the maximum value it stipulates, perform simplicity operation according to certain simplicity strategy;

Step 5: combine the corresponding component vectors of M individuals in the subspecies and all the T complete non-inferior solutions in the external file, and make calculation according to certain fitness function and choose M individuals as the final new generation of subspecies;

Step 6: determine whether the terminal conditions have been met, if met, the obtained external file is the optimum Pareto solution set of the algorithm; if not met, set the number of evolution generation plus 1 and transfer to step 2.

C. Analysis on the strategy and mechanism influencing the efficiency and solution space of the algorithm

1) Chromosome coding strategy

This algorithm adopts integer coding method to encode the individuals in the algorithm. First, sort the abstract member services in the efficient solutions of the abstract composite service, and the serial numbers of abstract member services are represented by integer element i; then sort the physical services in each abstract service and the serial numbers of the abstract services are represented by integer element j; the choice of physical services can be represented by a binary component (i,j). The chromosome codes of the efficient corresponding individuals of a certain physical composite service of task TD can be got as follows: $\{(1,2),(2,4),(3,2),\ldots,(i,j),(i+1,4),\ldots,(n,2)\}$, of which, n is the number of the abstract member services in the efficient solutions of the abstract composite service, and the chromosome codes include n binary genes, and the first element in the binary gene represents the only serial number of this gene. This chromosome coding strategy represents that the corresponding complete individuals of the efficient solutions of a physical composite service are composed of any one physical service option among each abstract services of the efficient solutions of abstract composite services.

2) Crossover operator

This algorithm adopts the direct crossover of the corresponding gene segment between individuals. First randomly create crossover positions i and j which are the series numbers of the genes in the corresponding chromosome codes, then directly change over the corresponding positions in the gene segment from i to j of the gene series number in the two individual chromosome codes, to generate two new individuals. If there are parent individuals $A_1=\{(1,1),(2,3),(3,2),(4,5),(5,3)\}$ and $A_2=\{(1,4),(2,5),(3,1),(4,7),(5,5)\}$, $i=3$, $j=4$, then the two new individuals $A`_1=\{(1,1),(2,3),(3,1),(4,7),(5,3)\}$ and $A`_2=\{(1,4),(2,5),(3,2),(4,5),(5,5)\}$ are obtained after crossover according to the crossover operator.

3) Mutation operator

This algorithm adopts simple mutation to realize the mutation of individuals. The method goes as follows: first, randomly produce mutation gene number i, and randomly choose a value in the available range of the corresponding variable values of the genes of this serial number to replace the original gene value. If there is parent individual $A_1=\{(1,4),(2,5),(3,8),(4,5),(5,6)\}$, the available range of the fourth gene variable is $R\in[1,21]$, first generating a random number $i=4$ and then generating the random number 14 in rang R, then we get the new after-mutation individual $A`_1=\{(1,4),(2,5),(3,8),(4,14),(5,6)\}$.

It is found experimentally that the cooperative evolutionary genetic algorithm will rapidly get the local optimum solution when the genetic mutation is adopted with the single low mutation probability. Therefore, this algorithm adopts two kinds of mutation probabilities to realize the mutation of individuals, among which the mutation probability of 0.6 is used when there is super individual and the mutation probability of 0.1 under common conditions.

4) The construction of fitness function

In the problem of multi-objective optimization, the fitness value of individual measures the performance of the solutions represented by the individuals. To estimate the fitness value of the individuals, a fitness function of individuals shall be constructed. The fitness functions of individuals of the optimization choice algorithm for physical composite service based on IMOCEGA can be divided into three types: fitness functions of sub-individuals, fitness functions of complete individuals and fitness functions with penalty variables.

(1)Standardization of QoS aggregation decision variables

The fitness functions of individuals of the optimization choice algorithm for physical composite service are constructed through the values of QoS aggregation vector of individuals, so firstly, the standardization conversion for the original QoS aggregation vector of individuals shall be made. Suppose if all individuals' QoS vectors V^T_{QoS} compose a decision matrix $A=(a_{ij})_{n\times m}$, of which, a_{ij} represents the value of the jth QoS aggregation decision variable of the ith individual, $max(a_{ij})$ represents the maximum value of the jth QoS aggregation decision variable of all individuals and $min(a_{ij})$ represents the minimum value of the jth QoS aggregation decision variable of all individuals, then we can get the standardization decision matrix of QoS aggregation decision vector V_{QoS}, which is $D=(d_{ij})_{n\times m}$, according to the formula 7.

$$d_{ij}=\begin{cases}\dfrac{a_{ij}-\min(a_{ij})}{\max(a_{ij})-\min(a_{ij})} & a_{ij}\in\{V^T_A,V^T_{SR},V^T_{\mathrm{R}},V^T_S\}\\ \dfrac{\max(a_{ij})-a_{ij}}{\max(a_{ij})-\min(a_{ij})} & a_{ij}\in\{V^T_P,V^T_{ST}\}\end{cases} \tag{7}$$

(2)Fitness function of sub-individuals

The core concept of multi-objective cooperative evolutionary genetic algorithm is to divide the multi-objective optimization problem into many sub-problems of single-objective optimization and make cooperative evolution for each single objective generating one subspecies. Therefore, this paper adopts all the single-objective functions in formula 6 as the fitness functions of sub-individuals of corresponding subspecies so as to solve the subspecies representative of each heredity.

(3)Fitness function of complete individual

The fitness function of complete individual adopts the weighing and construction of the QoS aggregation decision variables in QoS aggregation decision vectors.

The weight is a subjective variable and the minor change of its value will directly influence the difference of optimum solutions, and users are not the experts of weight setting and they hardly give the exact values respectively for the weights of all QoS decision variables, so the five levels of fuzzy literals *VH(Very High),H(High),M(Middle),L(Low),VL(Very Low)* are used for users to set the weights of QoS features. The exact values of weights is calculated as shown in formula 8, of which, W_P, W_{ST}, W_A, W_{SR}, W_R and W_S respectively are the weights of V_P, V_{ST}, V_A, V_{SR}, V_R and V_S of QoS decision variables.

$$\begin{aligned} & W_P, W_{ST}, W_A, W_{SR}, W_R, W_S \in \{VH, H, M, L, VL\} \\ & W_P + W_{ST} + W_A + W_{SR} + W_R + W_S = 1 \\ & \frac{H}{VH} = \frac{M}{H} = \frac{L}{M} = \frac{VL}{L} = \frac{3}{4} \end{aligned} \quad (8)$$

The fitness function of complete individual ftotal can be got by the formula 9 as shown.

$$f_{total} = W_P \times V_P + W_{ST} \times V_{ST} + W_A \times V_A + W_{SR} \times V_{SR} + W_R \times V_R + W_S \times V_S \quad (9)$$

(4)Fitness function with penalty variables

When cropping the external file, eliminate the optimum pareto solutions with small fitness to maintain the predetermined scale of external file. When keeping the excellent quality of individuals in the external file, guarantee the uniform distribution of individuals in the solution space to prevent the individuals in the external file from getting into the local optimality of solution space. This algorithm adopts the mean value of Hamming distance[15] to determine the similarity between individuals and take it as the penalty variables to adjust the fitness function of complete individual.

The Hamming distance *p(i,j)* between two individuals c_i and c_j can be solved with the formula 10, of which, $i,j=1,2,\ldots,m$, c_{ik} represents the value of the kth gene value of solution c_i and n represents the number of genes.

$$p(i,j) = \sqrt{\sum_{k=1}^{n} (c_{ik} - c_{jk})^2} \quad (10)$$

Then the mean value $\overline{p}(i)$ of Hamming distance of a certain individual c_i in the external file can be solved with formula 11, of which, m represents the number of individuals in the external file.

$$\overline{p}(i) = \frac{1}{m-1} \sum_{j=1, j \neq i}^{m} p(i,j) \quad (11)$$

The fitness function ftotal with penalty variables is established according to the formula 9 and formula 10, as shown in formula 12, of which, γ_{pen} is penalty coefficient.

$$f_{pen} = f_{total} + \gamma_{pen} \times \overline{p}(i) \quad (12)$$

(5)The mode of cooperation between subspecies and the estimation of individual fitness value

This algorithm adopts the random and greedy choice as the cooperation mode between subspecies to estimate the fitness value of individuals. First, combine the representatives of all other subspecies (which are the sub-individuals with the maximum fitness value solved from the fitness function of sub-individuals) and the sub-individual to be estimated in this subspecies to form a complete solution, and then randomly choose an individual from all other subspecies and combine it with the sub-individual to be estimated to form another complete solution candidate. If the former complete solution candidate can be Pareto-dominated by the latter solution for the total fitness function, reserve the solution candidate generated randomly; otherwise, reserve the solution candidate generated from the representatives.

(6)External files update

This algorithm adopts the complete solution candidate set generated from each generation of cooperative evolution to update the external files. Its updating process goes as follows: get all the complete solution candidate sets cooperatively generated from all subspecies, and if the solution candidates are not dominated by any existing solution candidate in external files, add it into the external file and delete all the existing solution candidates Pareto-dominated by this solution candidate; otherwise, the external file shall not be changed. The use of external file will quicken the algorithm convergence to the true Pareto optimum solution set of the problem.

(7)External file cropping and next generation of subspecies constructing

If the number of the solutions in the external file exceeds the stipulated maximum value, calculate the fitness values of all individuals in the external file and sort them and remove the solution with minimum fitness value. Construct the next generation of subspecies with the same method. The external file cropping and the construction mechanism of next generation of subspecies can make the optimum Pareto solutions of external file evenly distribute in the Pareto optimum leading edge of problem of multi-objective composite optimization.

(8)Termination criterion of algorithm

This algorithm adopts two types of termination criterions: one is to set the maximum value of number of evolution generations, and terminate the algorithm when the species evolves to the maximum number of generations; the other one is to set the update cycle *D* of the external file, if the external file has not changed during the continuous *D* times of evolution, we can conclude that the subsequent evolution can not perform much update on the external file and we can terminate the algorithm.

IV. SIMULATION EXPERIMENT AND ANALYSIS

The following will verify the effectiveness and feasibility of IMOCEGA in the optimization choice for the efficient solutions of QoS aggregation physical service. The comparison algorithm is traditional single-species genetic algorithm (GA). To reduce the complexity of the experiment, only consider the sequential model and reduce the decision variables to the three decision variables such as service price, service availability and service security, without changing the rationalization of the experiment. The hardware conditions for the experiment are: Pentium Dual Core

CPU 2.50GHz, memory 2GB; software conditions are: operating system Windows XP, programme software Matlab 7.

To compare the optimum Pareto solution values of IMOCEGA and GA and the convergence rate we can acquire at the same service composite scale, the common fitness function for both of IMOCEGA and GA must be determined. This paper adopts the gross mass of QoS based on weighted mean method shown in formula 9 as the fitness function of algorithm. Firstly, set the fuzzy weight *{VH,H,L}* and solve the weight of QoS features $W_P=0.615, W_A=0.307, W_S=0.078$ with formula 8, QoS decision variables V_P, V_A and V_S are generated from random numbers, and the fitness function of complete individual is shown in formula 13, which are used for the calculation of the fitness value of cooperative individuals in the IMOCEGA and the fitness value of single individual in GA; Set the penalty coefficient as 0.2.

$$f'_{total} = W_p \times V_P + W_A \times V_A + W_S \times V_S \quad (13)$$

The specific experiment parameters of this experiment are set as follows: besides the subspecies of multi-objective cooperative evolutionary genetic algorithm is set as 3, all the other parameters of the two comparative algorithms are the same, of which the population size is 50, the crossover probability is 0.7 and the mutation probabilities are 0.6 and 0.1, the number of genetic generations is 500, external file is 120 and the time unit is *s*. The experiment makes comparison for the statistical data from 50 times of operation, as shown in Table III, of which, the average maximum fitness value of GA is the mean value of the maximum fitness values obtained in the 50 times of operation, while the average maximum fitness value of IMOCEGA is the result of averaging the fitness values larger than the maximum fitness value of GA first obtained through IMOCEGA during the single operation for 50 times; the average time is the average value of the execution time when the maximum fitness value is obtained for the first time during the 50 times of operation; the average number of generations is the average value of operation generations number when the maximum fitness value is obtained for the first time during the 50 times of operation.

TABLE III.
THE COMPARATIVE EXPERIMENT RESULTS ON THE FITNESS FUNCTIONS WITH WEIGHTS

Algorithm comparison	Process number of composite service	Maximum Fitness Values	Average time	Average iteration
IMOCEGA	6	0.6574	42.98	75
GA		0.6496	39.36	156
IMOCEGA	15	0.6681	66.46	95
GA		0.6513	79.42	245
IMOCEGA	25	0.6757	133.94	124
GA		0.6593	184.11	340

Table III shows that when the process number of composite service is 6, the optimization choice algorithm for efficient solutions of the physical service based on IMOCEGA will obtain a fitness value which is larger than the fitness value obtained with the optimization choice algorithm based on GA. When the maximum fitness value is obtained, the average iteration number of IMOCEGA is apparently smaller than that of GA, which arises from that the number of the optimum Pareto solutions obtained by the IMOCEGA in each iteration process is several times of that of GA; the average execution time of IMOCEGA is little larger than that of GA, which arises from that IMOCEGA has cooperative interactive operation among subpopulations and fitness value estimation in each iteration while the GA does not, making the loop calculation time consumed by IMOCEGA for obtaining optimum Pareto solutions longer than that of GA. With the growth of the process numbers of composite service (when the process numbers of composite service are 15 and 25), the optimization choice algorithm for the efficient solutions of physical composite service based on IMOCEGA can obtain the fitness value larger than that of GA; meanwhile, both the average number of iteration times and the average optimization time of IMOCEGA are smaller than those of GA, which shows that the convergence speed for the efficient solutions of physical composite service obtained by IMOCEGA is more than that of the solutions obtained by GA.

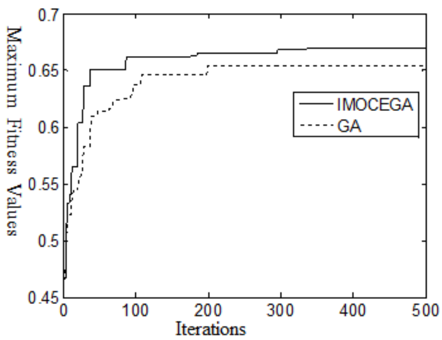

Figure2. The Evolution Cures (Maximum Fitness Values-Iterations) Comparison between the Optimum Solutions of IMOCEGA and GA

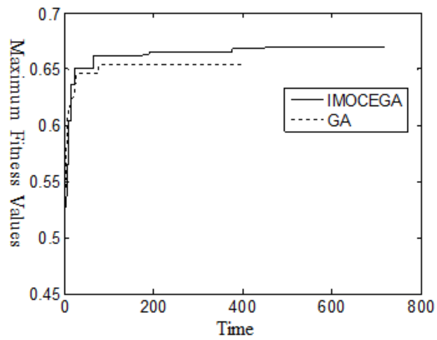

Figure3. The Evolution Cures (Maximum Fitness Values-Time) Comparison between the Optimum Solutions of IMOCEGA and GA

This paper has further given the linear optimization results obtained by using the algorithms of IMOCEGA and GA when the process number of composite service is 15, of which, the relationship between maximum fitness value and iterations is shown in Fig.2 and the relationship between maximum fitness value and calculation time is shown in Fig.3. We can conclude from Fig.2 that solving the problem of optimization choice for the efficient solutions of physical composite service by IMOCEGA can obtain more convergence speed and better fitness value. IMOCEGA takes more calculation time than GA during single evolution, so in Fig.3, the maximum fitness value of IMOCEGA within the equal time at the initial stage is little smaller than that of GA. However, in the overall operation time, IMOCEGA spends less time than GA in obtaining the same fitness values.

In fact, when solving the optimization choice problem of the efficient solution of physical composite service with GA, users must give the preference information on all QoS features, known as QoS objective weight vector; and one operation can get only one optimum solution, which is less efficient. Not giving QoS objective weight vector in advance, IMOGEGA can get many optimum Pareto solutions through Pareto dominance definition by one operation, extend the Pareto optimum solution space and effectively avoid the blindness of users to determine the QoS objective weight vectors.

V. CONCLUSION

Through the analysis on the result of the above comparison experiment, we can conclude that the multi-objective cooperative evolutionary genetic algorithm has provided an efficient and practical method for solving the optimization choice problem of the efficient solution of physical composite service in the complex control flow model, and it has more convergence speed and better solution capability in comparison with the traditional intelligent optimization algorithms.

ACKNOWLEDGMENT

This work was supported by The National Natural Science Foundation of China, Grant No. 60903174, the Scientific Research Plan Projects of Zhengzhou (Grant. 10PTGS508-3), Doctoral Research Foundation of Henan Institute of Engineering and Foundation.

REFERENCES

[1] Barichard V.. Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications. Lecture Notes in Economics and Mathematical Systems, 2009.

[2] Cao H.J., Jin H., Wu S., et al. ServiceFlow: QoS Based Service Composition in CGSP. In: Proceedings of the 10th IEEE International Conference on Enterprise Distributed Object Computing (EDOC2006). Hong Kong, China, 2006: 453-458.

[3] Li Zhen., Yang Fang Chun., Su Sen. Fuzzy Multi-Attribute Decision Making-Based Algorithm for Semantic Web Service Composition. Journal of Software. 2009, 20(3): 583-596.

[4] Fan Xiao Qin., Jiang Chang Jun., Wang Jun Li., Pang Shan Chen. Random-QoS-Aware Reliable Web Service Composition. Journal of Software. 2009, 20(3): 546-556.

[5] Canfora G., Penta M.D., Esposito R., et al. An approach for QoS-aware service composition based on genetic algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005), Washington DC, USA, 2005. 1069-1075.

[6] Jiang Zhe Yuan., Han Jiang Hong., Wang Zhao. An Optimization Model for Dynamic QoS-Aware Web Services Selection and Composition. Chinese Journal of Computers. 2009, 32(5): 1014-1025.

[7] Xia Hong., Li Zeng Zhi. A Particle Swarm Optimization Algorithm for Service Selection Problem Based on Quality of Service in Web Services Composition. Journal of Beijing University of Posts and Telecommunications. 2009, 32(4): 63-67.

[8] Zhu Qing., Li Guo Rong., Liu Guang Qiang., Wang Shan., Du Xiao Yong. Dynamic service composition algorithms in grid. Journal of Huazhong University of Science and Technology (Nature Science). 2006, 34: 134-137.

[9] Wang Chuang Wei., Qian Xue Zhong. Application of ant colony algorithm in web services composition problem. Computer Engineering and Design. 2007, 28(24): 5912-5914.

[10] Peng Xiao Ming., He Yan Xiang., Zhu Bing Jian. Application of Ant Colony Algorithm in Web Services Composition. Computer Engineering. 2009, 35(10): 182-187.

[11] Zhang Liang Jie., Li Bing., Chao Tian., et al. On Demand Web Services-Based Business Process Composition. In Proceedings of the IEEE International Conference on System, Man, and Cybemetics. Washington, USA, 2003, 4057-4064.

[12] Canfora G., Penta M D., Esposito R., et al. A lightweight approach for QoS-aware service composition. In Proc. 2nd International Conference on Service Oriented Computing, NewYork, USA, 2004, 36-47.

[13] Zhang Cheng Wen., Su Sen., Chen Jun Liang. Genetic Algorithm on Web Services Selection Supporting QoS. CHINESE JOURNAL OF COMPUTERS. Vol.29, No.7, July 2006.1029-1040.

[14] Jaeger M.C., Rojec-Goldmann G., Mühl G.. QoS Aggregation for Web Service Composition using Workflow Patterns. In: Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference, 2004: 149-159.

[15] Hamming., Richard W.. Error detecting and error correcting codes. Bell System Technical Journal, 1950, 26(2): 147–160.

Zhiwu. Wang, born in 1974, Ph.D., associate professor. His research interests include Grid computing, Cloud computing, Decision Support System.