An Improved Artificial Bee Colony Algorithm for Cloud Computing Service Composition

Bin Xu, Jin Qi, Kun Wang
School of Internet of Things
Nanjing University of Posts and Telecommunications
Nanjing, China
xubin.njupt@foxmail.com, qijin@njupt.edu.cn,
kwang@njupt.edu.cn

Ye Wang
College of Electronic Science and Engineering
Nanjing University of Posts and Telecommunications
Nanjing, China
Alisazhuang@foxmail.com

Xiaoxuan Hu, Yanfei Sun
School of Automation
Nanjing University of Posts and Telecommunications
Nanjing, China
comichu88@qq.com, sunyanfei@njupt.edu.cn

***Abstract*—The rapid increase of using cloud computing encourages service vendors to supply services with different features and provide them in a service pool. Service composition (SC) problem in cloud computing environment becomes a key issue because of the increase of service quantity and user requirements of the quality of service experience. To satisfy the demands on quality of service experience and realize an efficient algorithm for SC problem, a quality of experience (QoE) evaluation model based on fuzzy analytic hierarchy process (FAHP) for SC problem is put forward first. Then, an improved artificial bee colony (IABC) optimization algorithm for QoE based SC problem is proposed. The algorithm improves the updating mechanism of scout bees by introducing current global optimal solution to accelerate convergence velocity and eventually to improve the solution quality. Finally, the experimental results on QWS dataset show that IABC has a better performance on QoE based SC problem, compared with original ABC, PSO and DE.**

Keywords- Service Composition, Quality of Experience, Artificial Bee Colony

I. Introduction

The innovation of Internet technology especially cloud computing makes a large number of services existing in many real world applications. On the one hand, there are plenty of services which functions are similar or the same. On the other hand, the function of a single Service cannot meet the complicated demands of users. The main purpose of the service composition (SC) is to effectively combine single-functional services with small granularity distributed over the Internet on the basis of certain logic to form a new appreciation services and thus realize the re-utilization of services [8], which avoids the waste of resources and meets the needs of the users at the same time.

The QoS (Quality of Service) model is proposed to distinguish the services with similar functions and different non-functional properties. It is currently the most widely used service metrics. QoS of service includes the service execution time, cost, availability, reliability, credibility, etc. However, these indicators only give expression on the technical performance of services and ignore subjective factors of users; thereby it cannot directly describe the service recognition degree of users. Quality of Experience (QoE) is a kind of evaluation method which combines influence factors of the level of service, user and environment. Thus, it directly reflects the service recognition degree of users.

To satisfy the demands on quality of service experience and realize an efficient method for SC problem, a quality of experience (QoE) evaluation model based on fuzzy analytic hierarchy process (FAHP) for SC problem is put forward first. Then, an improved artificial bee colony (IABC) optimization algorithm for QoE based SC problem is proposed. In IABC, a new updating mechanism of scout bees by utilizing historical useful information is designed to speed up the convergence. Finally, the experimental results show that IABC has a better performance on QoE based SC problem, compared with original ABC, PSO and DE.

The rest of the paper is arranged as follows. Section II describes the related work. Section III gives the definition of QoE based SC problem. Section IV presents IABC designed for QoE based SC problem. Section V uses QWS dataset to testify the performance of IABC compared with the original ABC, PSO and DE. Finally, a conclusion is made in Section VI.

II. Related work

Solving SC problem successfully depends on two core issues. One is the appropriate quantitative description of the service quality, and the other is the method of service composition. In the traditional research, a service is usually expressed in multiple tuples with functional and non-functional properties, and the method for expressing the service quality quantitatively is the QoS mechanism. Since QoS focuses on the quality of service from the technical level

QSHINE 2015, August 19-20, Taipei, Taiwan

DOI 10.4108/eai.19-8-2015.2260856

and lacks of the presentation of user experience, the growing demand for service experience of users has not been satisfied. Therefore, it is essential to study the problem of service evaluation based on QoE. Currently, the research on QoE mainly includes the following aspects: (1) QoE definition, (2) QoE quantitative method, (3) QoE evaluation index and (4) QoE evaluation model. The international telecommunication union (ITU) defines QoE as the whole subjective acceptable level of the end user about the application or service [1]. Mean opinion score (MOS) [2] proposed by ITU is widely used to quantify the QoE at present [3]. Some works [4, 5] try to construct the correlation models of QoE based on QoS. In these works, the QoS indexes are preserved and transformed to the indexes in QoE model.

Since SC problem is a NP-hard [7], it is hard to obtain the optimal solution when the problem scale increases. Intelligent optimization algorithms are applied to SC problem recently. Yilmaz [6] utilized genetic algorithm to deal with SC problem, in which the heuristic simulated annealing and harmony search were introduced as intelligent mutation operators to fast the convergence speed. Wen [7] improved PSO to tackle SC problem, in which the division of circular orbits of particles is used to distinguish the discrete values of individuals.

The increase of the service experience requirements of users promotes us to study the following two questions: (1) How to introduce the QoE evaluation mechanism into SC problem? (2) How to improve the algorithm for SC to obtain a satisfied performance? To solve the above problems, this paper establishes a QoE evaluation mechanism of SC problem first. Then IABC is proposed to solve the QoE based SC problem.

III. PROBLEM DEFINITION

A. FAHP

Analytic Hierarchy Process (AHP) proposed by A.L. Saaty in the 1970s is a system analysis method which combines qualitative and quantitative analysis. AHP implements a five-step hierarchy ordering to calculate the combination weight of the constituent elements to obtain a comprehensive evaluation value of the different feasible schemes, which provides a basis for the selection of the optimal solution. FAHP combines AHP and fuzzy logic and reduces the influence of subjective factors on the evaluation in a certain degree. Following are some main concepts of FAHP [9].

Definition 1. Matrix $A=(a_{ij})_{n\times n}$ is the fuzzy matrix if it satisfies

$$0\le a_{ij}\le 1,(i=1,2,...,n;\ j=1,2,...,n) \qquad (1)$$

Definition 2. Fuzzy matrix $A=(a_{ij})_{n\times n}$ is the fuzzy complementary matrix if it satisfies

$$a_{ij}+a_{ji}=1,(i=1,2,...,n;\ j=1,2,...,n) \qquad (2)$$

Definition 3. Fuzzy complementary matrix $A=(a_{ij})_{n\times n}$ is the fuzzy consistent matrix if it satisfies

$$a_{ij}=a_{ik}-a_{jk}+0.5 \quad \forall i,j,k \qquad (3)$$

Fuzzy consistent matrix can synthesize fuzzy consistent matrixes given by many experts to form overall fuzzy consistent matrix thus forming an effective group decision making. TABLE I shows an example of the meaning of scale of 0.1 to 0.9 [10].

TABLE I. 0.9-0.1 SCALE

Scale	**State**
0.5	Two elements are equally important
0.6	One element is slightly more important than the other element
0.7	One element is more important than another element obviously
0.8	One element is much more important than the other elements
0.9	One element is extremely more important than the other element
0.1，0.2 0.3，0.4	It can be transferred to (1- current value). That is, for 0.1 (for element *a* to *b*), it is 1-0.1=0.9 (for element *b* to *a*).

According to the scale above, the judgment matrix $A=(a_{ij})_{n\times n}$, which consists of those scales above, has the following properties

$$a_{ij}+a_{ji}=1 \qquad (4)$$

$$a_{ii}=0.5 \qquad (5)$$

Theorem 1 Suppose $A=(a_{ij})_{n\times n}$ is fuzzy complementary matrix. The sum of row of *A* is

$$r_i=\sum_{k=1}^{n}a_{ik},i=1,2,...,n \qquad (6)$$

The matrix $R=(r_{ij})_{n\times n}$ is fuzzy consistent if

$$r_{ij}=\frac{r_i-r_j}{2(n-1)}+0.5 \qquad (7)$$

It has been demonstrated by [11] that the obtained fuzzy consistent matrix satisfies the consistence of human decision-making thinking and has a good robustness.

Theorem 2. Suppose that $A=(a_{ij})_{n\times n}$ is a fuzzy complementary matrix. The consistent element r_{ij} fuzzy consistent matrix $R=(r_{ij})_{n\times n}$ is calculated as follows.

$$r_{ij}=\frac{r_i-r_j}{2(n-1)}+0.5$$

where $r_i = \sum_{k=1}^{n} a_{ik^2}, i = 1,2,...,n$. Then, the weight vector $\omega = (\omega_1, \omega_2, ..., \omega_n)^T$ is computed as follows.

$$\omega_i = \frac{\sum_{j=1}^{n} a_{ij} + \frac{n}{2} + 1}{n(n-1)}, \quad i = 1,2,...,n \tag{8}$$

The details of the above theorems are shown in [11].

B. QoE model of Service

The evaluation method based on quantification QoE is different with the past objective method which just connects QoE with QoS by a simple equation. This method introduces the subjective feeling of experts and decision makers to ensure the selection of various indexes, the scientificity of giving weight and the consistency with the user perception. QoS parameters of service contain the following items: the cost of calling the service, response time, availability, throughput, production capacity, the success rate, reliability and so on. We select response time (*T*), availability (*A*) and reliability (*Rel*) , which are associated with Service QoE, as the influence indicators of QoE [12] thus establishing the QoE evaluation system structure of service, as shown in Fig. 1.

After establishing QoE evaluation system structure, the weight of every QoS parameter which is associated with QoE of service also needs to be determined [13].Two experts gives a comparison between the two index values and then establish a fuzzy complementary judgment matrix *A*.

$$A = \begin{array}{c} \\ t \\ a \\ r \end{array} \begin{array}{c} \begin{array}{ccc} t & a & r \end{array} \\ \begin{bmatrix} 0.5 & 0.2 & 0.4 \\ 0.8 & 0.5 & 0.7 \\ 0.6 & 0.3 & 0.5 \end{bmatrix} \end{array}$$

With method mentioned above, we can calculate the weight vector. ω =(0.26,0.42,0.32). Thus, the result of the service QoE evaluation model is shown in Fig. 1.

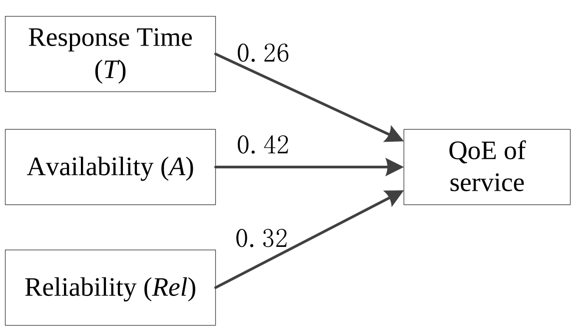


Figure 1. Service QoE evaluation model

C. QoE model of SC

Structured process is a very rigorous process model. It has a very regular behavior description and architecture and can be applied to most of the combination scenarios in the real world. Thus, this paper will combine structured process to discuss the process framework for service composition. Structured process has a unique starting point and end point and includes the four most common basic process structures: sequence structure, selective structure, parallel structure and circular structure [14]. So service composition model can also be divided into sequence structure, selective structure, parallel structure and circular structure. In practical application, most of composition services can be made by these four basic structures. Let's study the following four basic structures and analyze these structures and the corresponding QoE calculation model in detail. At the beginning and end of the process we are introducing two virtual nodes, which indicate the beginning and end respectively presented by *START* and *END*. The service modules used in the process is represented by *WS*.

1. Sequence structure: It indicates that the implement order of *WS* follows the sequence of *WS*. Execute the start node *START* first, then implement services from *WS1* to *WSn* according to its own sequence until reach the final end node *END*. The structure is shown in Fig. 2

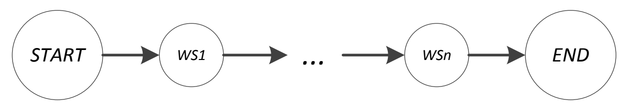


Figure 2. Sequence model

Each QoE parameter is calculated as follows.

$$\begin{cases} T_{seq}(WS_1 .. WS_n) = \sum_{i=1}^{n} T(WS_i) \\ A_{seq}(WS_1 .. WS_n) = \prod_{i=1}^{n} A(WS_i) \\ \mathrm{Re}l_{seq}(WS_1 .. WS_n) = \prod_{i=1}^{n} \mathrm{Re}l(WS_i) \end{cases} \tag{9}$$

2. Selective structure: It represents that a selected *WS* will be executed while other sets of services will not be executed. The structure is shown in Fig. 3, where *Pn* represents the probability that *WSn* is selected.

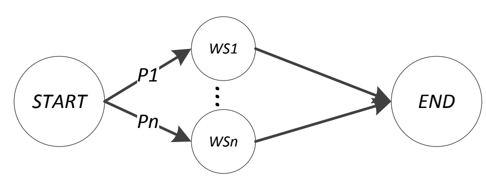


Figure 3. Selection model

Each QoE parameter is calculated as follows.

$$\begin{cases} T_{sel}(WS_1 \ldots WS_n) = \sum_{i=1}^{n} (pi * T(WS_i)) \\ A_{sel}(WS_1 \ldots WS_n) = \sum_{i=1}^{n} (pi * A(WS_i)) \\ Rel_{sel}(WS_1 \ldots WS_n) = \sum_{i=1}^{n} (pi * \mathrm{Re}\,l(WS_i)) \end{cases} \tag{10}$$

3. Parallel structure: It indicates that we choose n *WS* to implement at the same time. The next service will be triggered after all of these services are performed. The structure diagram is shown in Fig. 4.

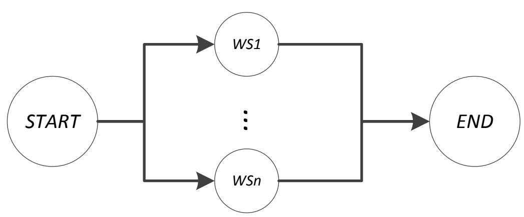


Figure 4. Parallel model

Each QoE parameter is calculated as follows.

$$\begin{cases} T_{par}(WS_1 \ldots WS_n) = Max(T(WS_1), T(WS_2) \ldots T(WS_n)) \\ A_{par}(WS_1 \ldots WS_n) = \prod_{i=1}^{n} A(WS_i) \\ Rel_{par}(WS_1 \ldots WS_n) = \prod_{i=1}^{n} \mathrm{Re}\,l(WS_i) \end{cases} \tag{11}$$

4. Circular structure: It shows that we do the loop execution *k* times on the sequence, from *WS1* to *WSn* . The structure diagram is shown in Fig. 5.

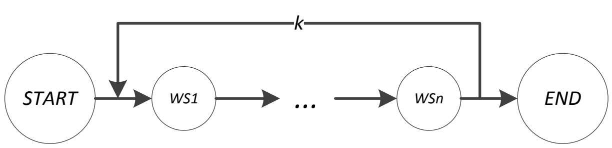


Figure 5. Circular model

Each QoE parameter is calculated as follows.

$$\begin{cases} T_{cir}(WS_1 \ldots WS_n) = \sum_{i=1}^{n} k_{cyc} * \sum_{i=1}^{n} T(WS_i) \\ A_{cir}(WS_1 \ldots WS_n) = (\prod_{i=1}^{n} A(WS_i))^k \\ Rel_{cir}(WS_1 \ldots WS_n) = (\prod_{i=1}^{n} \mathrm{Re}\,l(WS_i))^k \end{cases} \tag{12}$$

All QoE properties do not have comparability due to the different range and unit. The greater the response time is, the worse the service quality is, and the response time is called cost index, which belongs to the negative index. The greater availability and reliability are, the better the service quality is. These two indexes are efficiency indexes, which belong to the positive indicators. In order to unify calculation, it is necessary to standardize (normalize) QoE indexes. Here the range transformation [15] method is used to realize the standardization of QoE value of services.

Positive indicators (availability, reliability) are standardized as follows.

$$Q_{stand} = \begin{cases} \dfrac{Q - Q_{min}}{Q_{max} - Q_{min}}, & Q_{max} - Q_{min} \neq 0 \\ 1, & Q_{max} - Q_{min} = 0 \end{cases} \tag{13}$$

Negative indicators (response time) are standardized as follows.

$$Q_{stand} = \begin{cases} \dfrac{Q_{max} - Q}{Q_{max} - Q_{min}}, & Q_{max} - Q_{min} \neq 0 \\ 1, & Q_{max} - Q_{min} = 0 \end{cases} \tag{14}$$

where Q represents the value of T or A or Rel, Q_{stand} is the QoE attribute value after standardized.

For a service composition path, the QoE calculation equation is shown as (15).

$$f_{QoE} = \begin{cases} \alpha \sum_{i=1}^{n} T_i + \beta \prod_{i=1}^{n} A_i + \gamma \prod_{i=1}^{n} Rel_i & \text{Sequence} \\ \alpha Max(T_1, T_2, \cdots, T_n) + \beta \prod_{i=1}^{n} A_i + \gamma \prod_{i=1}^{n} Rel_i & \text{Parallel} \\ \alpha \sum_{i=1}^{n} T_i + \beta \prod_{i=1}^{n} p_i A_i + \gamma \prod_{i=1}^{n} p_i Rel_i & \text{Selective} \\ \alpha * l * \sum_{i=1}^{n} T_i + \beta (\prod_{i=1}^{n} A_i)^k + \gamma (\prod_{i=1}^{n} Rel_i)^k & \text{Circular} \end{cases} \tag{15}$$

where, T, A and Rel correspond to the standardized response time, availability and reliability of QoE values. α, β, γ represent the corresponding weights for the three properties in user experience quality evaluation, and $\alpha + \beta + \gamma = 1$. Thus, the objective of QoE based SC problem is to maximize f_{QoE}.

IV. IABC FOR QOE BASED SC PROBLEM

A. Basic Principles of ABC Algorithm

ABC proposed by Karaboga [16] in 2005 is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm. The algorithm attracts the attention of many scholars, because of its fast convergence, less control parameters, and easy to implement.

According to the division of labor, the artificial bee colony in ABC is divided into three types: employed bees, onlookers and scouts. They cooperate and communicate with each other, which makes the evolution of the work of collecting nectar to the efficient direction. The employed bees are to find high quality honey and implement preliminarily neighborhood search. Onlookers wait on the information of nectar quality from employed bees for making decision to select food source for neighborhood search. The food source that has higher quality (fitness) will have a large chance to be searched randomly by the scout bees. Each bee corresponds to a solution and employed bees represent the existing solutions which constitute the current population; onlookers represent potential neighborhood search solutions, which have access to the population to become existing solution; scouts represents the global random search solutions which can replace the obsolete existing solutions.

In ABC, the position of a food source represents a possible solution of the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of employed bees is equal to the number of onlookers and the number of solutions in a population. First of all, ABC algorithm generates the initial population with *SN* solutions (food sources).Each solution is a *D*-dimensional vector. Then the bees executes loop search on all food sources. The employed bees conduct a neighborhood search on corresponding food sources (solutions) firstly and select the food sources with high quantity of nectar that is higher fitness. After all employed bees have completed the search, they share the nectar information of the food sources with the onlooker bees on the dance area and onlooker bees choose the food source according to information obtained in accordance with the probability. The more nectar food sources, the greater probability of being selected by the onlooker bees. Later, the onlooker bees are also on a neighborhood search, and choose a better solution.

An onlooker bee chooses a food source depending on the probability P_i which is associated with the food source and calculated by (16).

$$P_i = \frac{fitness_i}{\sum_{n=1}^{SN} f_n} \tag{16}$$

Where the $fitness_i$ is the fitness value of the solution *i*.

The employed bees and onlookers conduct neighborhood search based on (17).

$$v_{ij} = x_{ij} + rand(x_{ij} - x_{kj}) \tag{17}$$

where $k \in \{1, 2, ...SN\}, j \in \{1, 2, ...D\}$, the k and j are randomly chosen indexes. Although *k* is determined randomly, it has to be different from *i*. The *rand* is a random number between [−1, 1].

If a solution has not been improved after limit cycles, the solution will be abandoned. "*Limit*" is an important control parameter in the ABC algorithm. The solution x_i is assumed lost, and then the scouts will generate a new solution instead using (18).

$$x_i^j = x_{\min}^j + rand(0,1)(x_{\max}^j - x_{\min}^j) \tag{18}$$

where x_i^j represents the *j*-dimensional solution vector component. $x_{\max}^j$ and $x_{\min}^j$ respectively represents the upper and lower limit of the *j*-dimension component of solution vector, and rand (0,1) is the uniformly distributed random number in the interval (0,1).

B. *IABC*

1. Adaptive parameter adjustment of limit

In the original ABC, generally *limit* is a fixed value. The researchers demonstrated that if you want to update all the parameters of a solution, the loop need to be implement at least *D* times. The *limit* is generally set to *SN* × *D* more appropriate because ABC updates only one dimension of the solution in a sufficiently large *MCN* (maximum number of iterations). However, aimed at solving the QoE based SC problem in this paper, the algorithm *MCN* should not be too large and generally is 100. Otherwise, the performance of the original algorithm cannot be improved. In the case of *SN* = 100, *D* = 10, 20, 30, theoretical value of the limit is larger than *MCN*, which causes the *limit* value extremely large and scouts completely lose its function, so that the global optimization capability of ABC is weakened. On the contrary, the smaller value of the *limit* will produce excessive scouts, which leads to the reduction of local searching ability.

To solve these problems, this paper improves the setting of the *limit* value. According to the current number of iterations *iter* and the maximum number of iterations *MCN*, an adaptive limit dynamic adjustment process is generated [17]. The calculation equation is as follows:

$$limit = [\frac{MCN}{iter}] \tag{19}$$

where [] is rounding symbol, *iter* counts from 1. At the early stage of the iterative algorithm, (19) controls the *limit* in a relatively large value and enhances the ability of local search algorithms. And in the middle and later period, the *limit* parameter is controlled in a relatively small value in order to prevent premature convergence of the algorithm. Once the solution corresponding to the current employed bees has not been significantly improved after iterating limit times, then the employed bees immediately convert to scouts, which is to re-initialization to improve the global search ability. Experiments bellow show that adaptive-adjustment *limit* argument always brings obvious improvement effect and enhances the universality of the algorithm whether *D* = 10, *D* = 20 or *D* = 30.

2. Global optimal guidance

In ABC, if the quality of nectar has not updated after iterating limit times, then the corresponding lead bee scouts will be converted to the onlookers by (18) to generate a new solution. But the fitness of new solution generated randomly is generally not high and will soon be eliminated in following selection process of onlookers. In the process of iteration, the global optimal solution x_{best} records the historical useful information. It can be used to generate new solutions to improve the quality of them. Thus, instead of (18), an improved solution generating method of scout is

$$x_i^j = \begin{cases} x_{best}^j & , j \neq r(\frac{i}{D}) \\ x_{\min}^j + rand(0,1)(x_{\max}^j - x_{\min}^j), & j = r(\frac{i}{D}) \end{cases} \tag{20}$$

where the operator $r(i/D)$ represents i represents the remainder of i mod D. x_{best}^j represents the value of the j dimension of the current global optimal solution.

After a certain number of iterations, there will be many nectar groups facing elimination because they are caught in local optimum. On the one hand, we have introduced the current global optimal solution x_{best} which produces a new solution to induce the bees to explore in the next iteration to a better solution. On the other hand, we also noticed that if only assign x_{best} to scouts then the diversity of population will decrease greatly. The solutions which face to eliminate in next iteration is initialized as x_{best}. Thus, we proposed a method to improve the scout bees initialization by judging whether the dimension j of the current solution is equal to the remainder of i mod D. If equal, then the corresponding value of the dimension is changed to the corresponding value of the corresponding dimension of x_{best}. On the contrary, the dimension has been initialized randomly. To sum up, the improved initialization method of scout uses the useful information in the x_{best} to guide scouts to the direction, accelerate the update efficiency of solution. While the introduction of random disturbance increases the diversity of population, so as to avoid trapping in local optimum.

C. IABC solving the problem of service composition

To apply IABC in QoE based SC problem, it is important to encode the solution appropriately. The service composition problem of sequence structure altogether contains n subtasks. Assume that the number of candidate services for each subtask is m and each candidate subtask service is represented by WS_{ij}, where i is the serial number of subtasks and j is the number of sub services to carry out the subtask. Each member of the bee swarm represents a path of service composition expressed with vector. The dimension of solution is same as the number of subtasks. For example, the service composition path $WS_{13} \to WS_{25} \to WS_{30} \to WS_{47} \to ...WS_{n4}$ and $WS_{17} \to WS_{24} \to WS_{32} \to WS_{49} \to ...WS_{n1}$ can be represented by X=[3,5,0,7,...4] and X=[7,4,2,9,...1] respectively. Thus, the corresponding solution is an integer vector.

The algorithm will update the coding of existing solution only when the fitness value of solution has been improved. The update mechanism is as follows.

$$v_{ij} = x_{ij} + [rand(0,1)(x_{ij} - x_{kj}) + 0.5] \tag{21}$$

where [] is the integral symbol. By the same token, the initialization method is changed to

$$x_i^j = \begin{cases} x_{best}^j, j \neq r(\frac{i}{D}) \\ x_{\min}^j + [rand(0,1)(x_{\max}^j - x_{\min}^j) + 0.5], j = r(\frac{i}{D}) \end{cases} \tag{22}$$

The steps of IABC for QoE based SC problem are described as follows.

Step 1. Set algorithm parameters. The number of employed bees = the number of onlookers = *SN*, maximum number of iterations is *MCN* and, the control parameter is *limit*.

Step 2. Initialize swarm. *SN* nectar sources are randomly generated and calculate the fitness function on the basis of the service portfolio topologies and (15).

Step 3. Employed bee conducts neighborhood search according to (21) and adopts the "greedy principle" to select according to the fitness value of old and new honey sources.

Step 4. Calculate the probability that each nectar source is chosen according to (16), onlookers use the principle of "roulette wheel" to select and execute the neighborhood search according to (21).

Step 5. Record the current global optimal solution x_{best}.

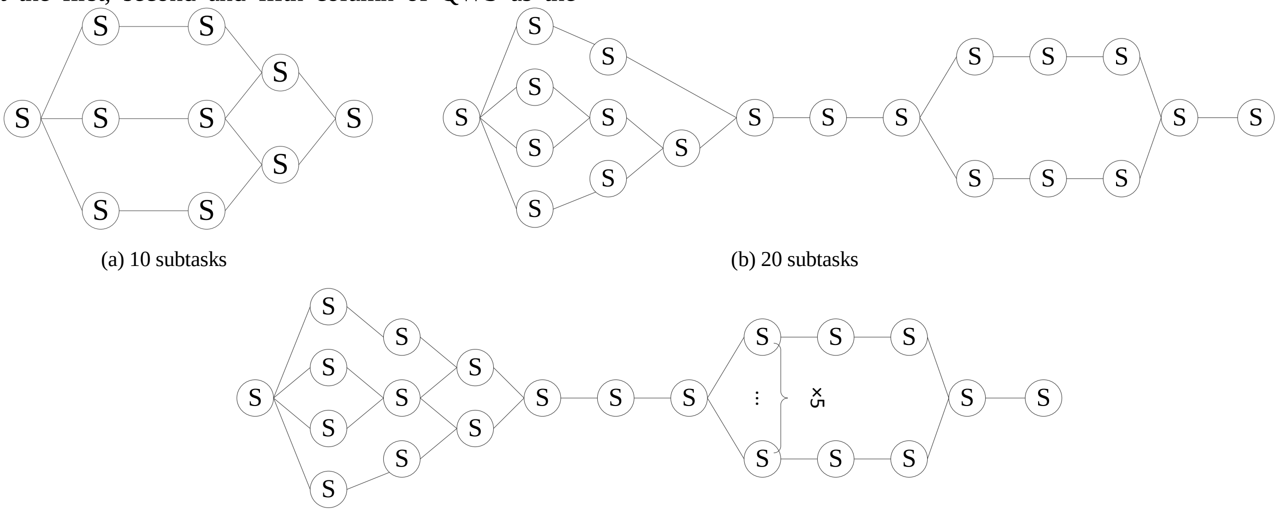
Step 6. *Limit* parameter vary adaptively with the number of iterations. If the fitness value of a nectar source has not been improved after *limit* iteration, the corresponding employed bees turn to scouts. The new nectar sources are generated according to (22).

Step 7. Record the presently optimal nectar source. If the current iteration number is less than MCN, turn to step 3 for the next iteration. Otherwise, output optimal solution.

V. EXPERIMENTS AND ANALYSIS

In this section, we adopt three typical Directed Acyclic Graphs (DAGs) [18] shown in Fig. 6 as task graphs of QoE based SC problem to verify the performance of IABC. Since the total number of subtasks is usually no more than 50, and it is the huge number of candidate services corresponding to different subtasks that expand the solution space of SC problem, so we take the three cases which respectively includes 10, 20 and 30 tasks as test cases. These three topology graphs include sequential, selective, parallel, and hybrid structure. Meanwhile, the quantity of candidate service for each subtask is set to be 100, therefore the solution space ranges is $100^{10} - 100^{30}$. Considering the scale of SC in reality, our testing scales can meet the requirements.

The testing dataset of the experiment is QWS (Al-Masri&Mahmoud, 2009) [19], which includes various parameter properties of 2507 actually existing web services, such as response time, availability, throughput, reliability, reputation, etc. Since the presented QoE based SC problem model only involves response time, availability and reliability, we select the first, second and fifth column of QWS as the testing dataset. In addition, we pre-process the corresponding data while selecting each column. Once the response time is bigger than 1000ms and the availability or the reliability is less than 50 percent, this service should be abandoned. For those services which meet the constraints, their properties will be standardized according to (13), (14).



(a) 10 subtasks

(b) 20 subtasks

(c) 30 subtasks

Figure 6. Three adopted structures (a) 10 subtasks (b) 20 subtasks (c) 30 subtasks

The max iteration *MCN* is 100, the population size *SN* is 100 and the initial value of "*limit*" is 100 and weights of *T*, *A*, and *Rel* are set to 0.26, 0.42 and 0.32 respectively according to the calculating method above. Considering the randomness of intelligent algorithm, the experiment is set to run 100 times and maximum, minimum, mean and median of the results are recorded. In order to prove the superiority of the improved algorithm, we choose original ABC, PSO and DE for comparison, the individual encoding method of the three algorithms is the same as that of IABC, the parameters are given by TABLE Ⅱ.

TABLE II. PARAMETERS OF THREE ALGORITHMS

Algorithm	**Parameters**
ABC	SN is 100, *limit* is 100,*MCN* is 100
DE	Population size is 100, mutation strategy is DE/rand/1, scaling factor of differential vector is 0.5, crossover probability is 0.5.
PSO	Population size is 100, inertia weight is w=0.9-(0.9-0.4)*iter/maxiter, cognition coefficients are c1=c2=1.7

Fig. 7 shows the average fitness values obtained from original ABC, PSO, DE and IABC with 100 runs which change along with iterations increase. It can be observed that the performance of IABC is better than the other algorithms in all three cases. In earlier iteration, IABC can rapidly improve the fitness of the solution to a high value, which benefits from the new solution generating method of scout utilizing the historical useful information. In later period, the fitness value of IABC reaches a flat state earlier which means the convergence speed is faster. Especially when the solution space gets larger, the difference of fitness among IABC and other 3 algorithms becomes more obvious. These show that IABC is more suitable for solving complex SC problem.

Fig. 8 shows contrast results of the 4 algorithms running 100 times, including maximum, minimum, mean and median of the fitness value. Comparing three bar chats (a), (b) and (c), under the same circumstance, all the results of IABC are comprehensively superior to original ABC, PSO, and DE. Furthermore, the maximum, minimum, mean and median values are very close, which indicates IABC has better stability and adaptability for varies problems.

VI. CONCLUSION

This paper establishes evaluation models of QoE based on FAHP and then gives a mathematical model of SC problem. By introducing global optimal solution to guide scout searching behavior and adaptive change mechanism of “limit”, we propose an IABC and apply it to QoE based SC problem. The simulation results show the good performance of IABC. Though QoE based SC model can express the experience of experts or decision makers, it is still need to be further improved in order to better describe the subjective feelings of users. These will be the emphasis of our further work.

ACKNOWLEDGMENT

This paper was supported by National Natural Science Foundation of China (61401225), China Postdoctoral Science Foundation funded project (2015M571790), NUPTSF (Grant No. NY213047, NY213050, NY214102 and NY214098).

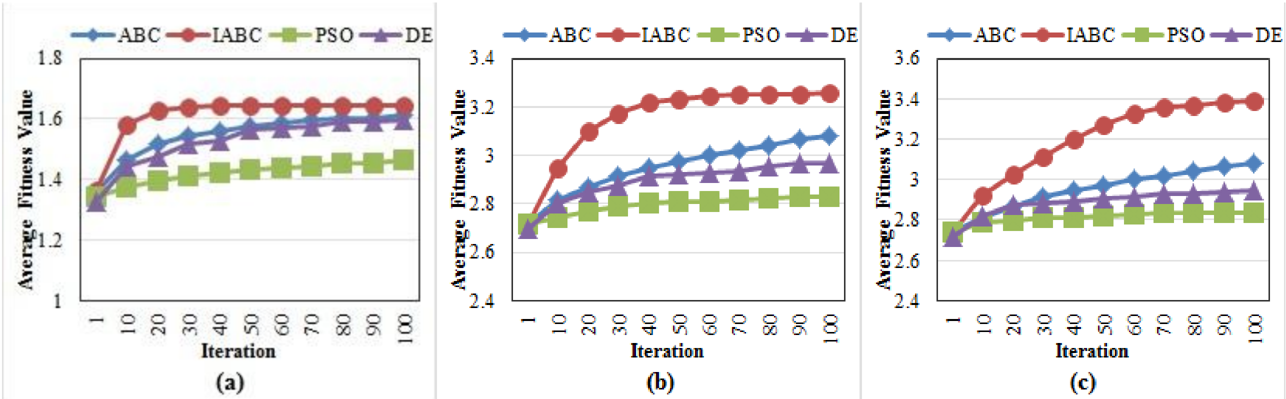


Figure 7. Convergence of 4 algorithms. (a) 10 subtasks, (b) 20 subtasks, (c) 30 subtasks

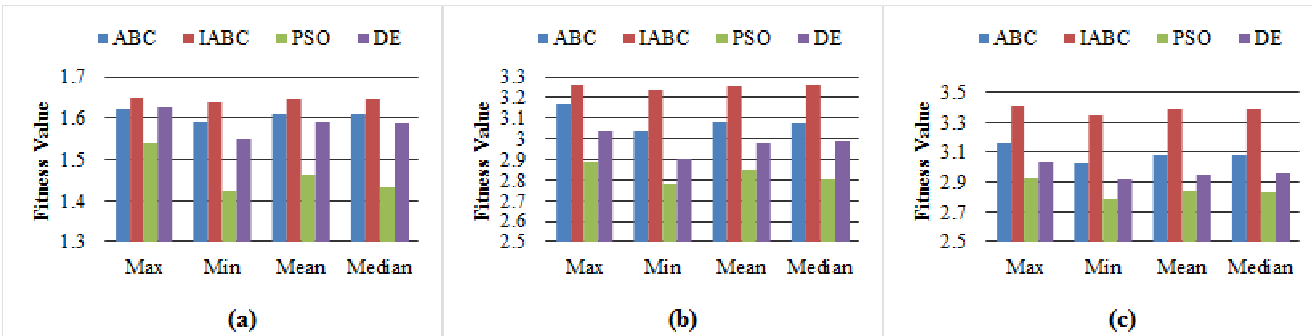


Figure 8. The statistical results of the 4 algorithms of 100 runs. (a) 10 subtasks, (b) 20 subtasks, (c) 30 subtasks

REFERENCES

[1] K. Kilkki, "Quality of Experience in Communications Ecosystem," J. UCS, vol. 14, pp. 615-624, 2008.

[2] J. G. Beerends, A. P. Hekstra, A. W. Rix, and M. P. Hollier, "Perceptual evaluation of speech quality (pesq) the new itu standard for end-to-end speech quality assessment part ii: psychoacoustic model," Journal of the Audio Engineering Society, vol. 50, pp. 765-778, 2002.

[3] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "QoE-driven cache management for HTTP adaptive bit rate streaming over wireless networks," IEEE Transactions on Multimedia, vol. 15, pp. 1431-1445, 2013.

[4] H.-J. Kim, D. H. Lee, J. M. Lee, K.-H. Lee, W. Lyu, and S.-G. Choi, "The QoE evaluation method through the QoS-QoE correlation model," in Fourth International Conference on Networked Computing and Advanced Information Management, pp. 719-725, 2008.

[5] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," Network IEEE, vol. 24, pp. 36-41, 2010.

[6] A. E. Yilmaz, and P. Karagoz, "Improved Genetic Algorithm Based Approach for QoS Aware Service Composition," in IEEE International Conference on Services (ICWS), pp. 463-470, 2014.

[7] T. Wen, G.-j. Sheng, Q. Guo, and Y.-q. Li, "Service composition based on modified particle swarm optimization," Chinese Journal of Computers, vol. 36, pp. 1031-1046, 2013.

[8] M. Alrifai, T. Risse, and W. Nejdl, "A hybrid approach for efficient Service composition with end-to-end QoS constraints," ACM Transactions on the Web (TWEB), vol. 6, pp. 7, 2012.

[9] T. L. Saaty, and L. G. Vargas, Models, methods, concepts & applications of the analytic hierarchy process: Springer Science & Business Media, vol. 175. 2012.

[10] Y. Du, W.-a. Zhou, and B.-f. Chen, "A QoE based evaluation of service quality in wireless communication network," in International Conference on New Trends in Information and Service Science, pp. 552-557, 2009.

[11] M. Yao, and Y. Huang, "Research on methodology of fuzzy decision making," Systems Engineering-Theory & Parctice, vol. 19, pp. 61-64, 1999.

[12] J. Pokhrel, F. Lalanne, A. Cavalli, and W. Mallouli, "QoE estimation for service selection using a Fuzzy-Rough hybrid expert system," in IEEE 28th International Conference on Advanced Information Networking and Applications (AINA), pp. 629-634, 2014.

[13] H. J. Kim, and S. G. Choi, "A study on a QoS/QoE correlation model for QoE evaluation on IPTV service," in The 12th International Conference on Advanced Communication Technology (ICACT) pp. 1377-1382, 2010.

[14] F. Tao, D. Zhao, H. Yefa, and Z. Zhou, "Correlation-aware resource service composition and optimal-selection in manufacturing grid," European Journal of Operational Research, vol. 201, pp. 129-143, 2010.

[15] Y. Chen, and G. Chen, "Classification & research advancement of comprehensive evaluation methods," Journal of Management Sciences in China , 2004.

[16] D. Karaboga, and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," Journal of global optimization, vol. 39, pp. 459-471, 2007.

[17] K. J. Åström, and B. Wittenmark, Adaptive control: Courier Corporation. 2013.

[18] F. Tao, Y. LaiLi, L. Xu, and L. Zhang, "FC-PACO-RM: a parallel method for service composition optimal-selection in cloud manufacturing system," IEEE Transactions on Industrial Informatics, vol. 9, pp. 2023-2033, 2013.

[19] http: //www.uoguelph.ca/~qmahmoud/qws/index.html.