

A Diffraction Service Composition Approach Based on S-ABCPC: An Improved ABC Algorithm

Xunyou Min, Harbin Institute of Technology, China*

Xiaofei Xu, Harbin Institute of Technology, China

Zhongjie Wang, Harbin Institute of Technology, China

Zhizhong Liu, Yantai University, China

ABSTRACT

In recent years, research on the QoS-aware service composition problem often assumes that each component service in the process to be solved is equally essential. They do not consider the impact of core component services and other component services on problem-solving, or even though their impact is considered, they are not fully considered. So this paper first proposes a diffractive method based on them. Considering the advantages of artificial bee colony (ABC) such as simplicity, this paper chooses it as the basic algorithm. In addition, with the continuous development of service ecosystem, it gradually formed a variety of domain features. They have an important influence on problem-solving, but the existing research has not explored this influence in-depth. Therefore, this paper digs deep into this influence. Given the characteristics of the problem to be solved in this paper, the S-ABCPC algorithm is designed. At last, experiments have proved the effectiveness of the method proposed in this paper. The impact factors of this method have been studied.

KEYWORDS

Artificial Bee Colony (ABC), Core Component Services, Diffractive Method, Domain Features, Quality of Service (QoS), S-ABCPC Algorithm, Service Composition, Service Ecosystem

1. INTRODUCTION

With the development of service-oriented computing, the real needs put forward by users are no longer simple. So basic services with a single function can hardly meet this demand. Service composition technology emerged under this background. A typical problem in service composition is the binding of each abstract component service in a process to a concrete Web service in the set of candidate services so that the binding result can satisfy a series of local qos constraints and global qos constraints, and the optimal solution is obtained. This typical problem is qos-aware service composition(Zeng

DOI: 10.4018/IJISMD.300778

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

et al., 2004). Many achievements have been made in the research on this problem, however, these studies identify that all component services in a service composition process are equally important. However, there are usually some component services that are core component services, that is to say, the quality of their binding results has a decisive and more significant impact on the quality of the overall combination result. The set formed by these core component services is called the core component service set. This is simply called the core service set. In addition, we note that due to the particularity of core component services in the service composition process, there will be some component services that can assist core component services and are closely related to them. This paper refers to these component services closely related to core component services as diffraction associated component services. The set formed by the corresponding diffraction associated component services is called the diffraction associated set. The core service set and the diffraction associated set have an essential influence on the solving of the service composition problem. For example, the core service set plays a leading role in the process of service composition, the performance of the core component services determine the performance of the entire composite service, the core service set and the diffraction associated set have an effect on improving the qos of the final solution result. In the previous related researches, when proposing methods to solve the service composition problem, the influence of core service set and diffraction associated set on problem-solving is often not considered, or even though their impact is considered, they are not considered in-depth and then used in a deeper level. To solve the above problem, this paper proposes a diffractive service composition method. This method adopts the processing method of distinguishing the primary and secondary and splitting the process to solve the problem. Specifically, this method first combines core component services, their corresponding diffraction associated component services and other corresponding component services using the corresponding algorithm for the first stage of combination. Secondly, uses the combined result of the first stage as the reference point, and uses the corresponding algorithm to combine it with the corresponding diffraction associated component services and other corresponding component services for the second stage of combination. After that, and so on until the combined result of the entire process is obtained.

The above answers the question “which solution method and strategy should be adopted in order to solve the composition problem” in this paper. Then based on the guidance of the method and strategy, use some suitable targeted algorithm to solve. So another question needs to be considered, that is, “now that the diffractive service composition method is adopted, what algorithm should we choose for solving the problem?” Considering the advantages of the ABC, and after our previous research, we found there is a strong correspondence and mapping between the mechanism of the optimization process of ABC and the solution process of service composition. For example, the behavior of honeybee colonies searching for the optimal food source corresponds to the process of finding the optimal solution to the composition problem, food sources correspond to feasible solutions in composition problems, the calculation of the fitness function corresponds to the calculation of the objective function of the composition problem, the end condition of the ABC corresponds to the end condition of the composition problem. Therefore, this paper chooses the ABC as the solution algorithm to solve the service composition problem. At present, some scholars have used ABC to solve the service composition problem(He et al., 2013) (Wang et al., 2013) (Chifu et al., 2010). However, these ABC-based solutions have some shortcomings, they often fail to consider the influence of domain features on the problem-solving process, or the effect of multiple domain features on the solution of the problem is not considered simultaneously, so the solution to the problem is out of touch with the actual situation in the service application, making the efficiency and effect of solving the problem not ideal. In fact, with the continuous development and evolution of service applications and service ecosystems involved in various service industries, gradually show the unique domain features in the service industry(such as priori feature, correlation feature, similarity feature and so on), they have an important influence on problem-solving. So this paper conducts an in-depth analysis of the optimization and running mechanism of ABC, mining the influence of multiple service domain

features on the problem-solving process to guide the solution of the problem, to further improve the efficiency and effect of problem-solving based on the diffractive service composition method. Given the characteristics of the composition problem solved in this paper, this paper designs a priori correlation ABC algorithm.

In short, this paper proposes a diffractive service composition method, first, the identification method of the core component services and the diffraction-associated component services corresponding to each diffraction combination stage is proposed, and then the process is divided diffractively. Secondly, perform constraint decomposition according to the constraint decomposition method proposed in this paper, decompose the global constraints proposed by the user into sub-constraints corresponding to each diffractive combination stage. At last, based on the sub-constraint of each diffractive combination stage and the corresponding sub-process, use a priori correlation ABC algorithm to solve until the final result that can meet the needs of the user is obtained.

Section 2 reviews related work. Section 3 introduces the SL-GABA algorithm. Section 4 proposes a diffractive service composition method, including the identification method of the core component services and the diffraction-associated component services of each layer, the constraint decomposition method and the priori correlation ABC algorithm. Section 5 shows the experimental results. Finally, the conclusion and future work are presented.

2. RELATED WORK

In the related research fields of cloud computing and service computing, Qos-aware service composition is one of the important problems, many scholars are committed to research on this problem. Literature (Khanouche et al., 2020) proposed a new qos-aware service composition method (FQSC method) that can improve the feasibility and solution time of the combination while maintaining the quality of the solution. Literature (Salam et al., 2019) proposed a multi-stage method to solve service composition, in the first stage QoS attributes are preprocessed, the second stage operates on the service clustering of the first stage, the third stage performs the service composition operation. In this paper, the NSGA-II algorithm is also integrated into the MapReduce framework, and through clustering operations to narrow the space that needs to be searched. Literature (Wang, Gu, Yu et al, 2019) proposed an adaptive service composition method based on deep reinforcement learning (DRL), this method is especially suitable for partially observable application environments, and the effectiveness, scalability and adaptability of this method are proved through experiments. Literature (Wang et al., 2020) converts the automatic service composition problem into an automatic planning problem and proposes a Q-Graphplan method, this method can avoid redundancy and improve search speed when generating planning diagrams, finally, experiments are carried out on the data set WSC-2009 to verify the excellent performance of this method. Literature (Elfirdoussi & Jarir, 2019) proposed a dynamic service composition framework (DWSC), this framework is mainly implemented based on two components: service dynamic designer (WSDD) and service selector (WSSP), this framework automates service composition based on popularity, finally, an experiment was conducted on the actual case of the transportation process and confirmed the effectiveness of this framework. Literature (Yaghoubi & Maroosi, 2020) proposed a new service composition optimization algorithm (IMVO) to improve the QoS of the final solution while ensuring that the service level agreement (SLA) is met, finally, it is verified through experiments that the algorithm can improve the QoS of the final solution. Literature (Li et al., 2019) solves the problem of service composition through graph database, the shortest bidirectional breadth-first algorithm and Dijkstra algorithm are used to solve the problem, preprocesses the service composition and stores them as paths in the directed bipartite graph in the graph database, finally, the correctness and performance of this method are verified through experiments. Literature (Niu et al., 2019) considers the uncertain QoS of the services, the uncertain QoS aware service composition problem (WSC) is modeled, turns this problem into a corresponding multi-objective optimization problem, and uses a non-deterministic multi-objective evolutionary

algorithm to solve, finally, a lot of experiments were carried out on the simulated data set, experimental results prove the superiority of this method. Literature (Haihong et al., 2019) provides a visual service composition system, introduces an asynchronous mechanism based on message subscription, and can quickly publish results. Some researchers use swarm intelligence algorithms (evolutionary algorithms) to solve qos-aware service composition. Literature (Yang et al., 2019) proposes a hybrid ant colony genetic algorithm (DAAGA), a fusion evaluation strategy is adopted to dynamically adjust the number of calls of genetic algorithm and ant colony algorithm, and iteratively adjusts the threshold to dynamically adjust the genetic operation and population size, thereby improving the accuracy and stability of service composition solution. Literature (Zhou et al., 2019) studies the recent evolutionary algorithm for multi-objective service composition (EMaO), the searching ability of these algorithms is studied experimentally, and this paper provides relevant suggestions for other researches on how to choose the appropriate EMaO algorithm. Literature (Peng et al., 2020) proposes a multi-cluster adaptive brainstorming optimization service composition algorithm (MCaBSO), this algorithm searches in a reduced solution space, this algorithm can perform an efficient search in a reduced solution space and can obtain higher quality solutions, finally, two data sets are used to verify the effectiveness and efficiency of this method. Literature (Fekih et al., 2019) proposes a harmony particle swarm optimization algorithm (HPSO), this method includes two filtering operations: the skyline operation and the local consistency enhancement operation, they reduce the search space and retain the most suitable alternative services, finally, the experimental results prove the effectiveness of this method. Most existing work believes that the QoS of web services is basically unchanged, however, unexpected situations may occur during the execution phase of the service composition, therefore, the literature (Wang, Ma, Chen et al, 2019) proposes a robust service composition method, this method can make the service composition maintain a high quality in the execution phase, finally, the experiment proves that this method can outperform the corresponding comparison algorithms. Literature (Thangaraj & Balasubramanie, 2020) combines hybrid meta-heuristic genetic algorithm and tabu search to perform service composition, experimental results show that high reliability, maximum throughput and interoperability can be obtained. Taking into account the advantages of the bee colony algorithm, some researchers use this algorithm to solve the qos-aware service composition. Reference (Zanbouri & Jafari Navimipour, 2020) chooses the bee mating algorithm, and combined with the trust-based clustering algorithm to solve the trust crisis in the service composition, simulation experiments show that the method in this paper is better than particle swarm algorithm, genetic algorithm and other comparison algorithms in solving small-scale problems, but the effect is not good enough to solve large-scale problems. Literature (Seghir et al., 2019) proposes an interval-based multi-objective artificial bee colony algorithm (IM_ABC) to solve the service composition problem, an interval-based feasibility analysis method is proposed for the interval constraint problem, finally, corresponding experiments are done based on real data and randomly generated data to prove the effectiveness and superiority of the method proposed in this paper. However, these research works did not consider the important influence of core component services and domain features on problem-solving, so it affects the performance of problem-solving. A few researchers have proposed methods to improve problem-solving by using core component services and domain features. Literature (Ma et al., 2014) gives the definition of core service set, and on this basis, a service composition method considering the core service set is proposed, however, this paper only reflects the impact of core service set on service composition by changing the weights in the aggregation function, the influence of the core service set on the service composition solution process and the corresponding service composition solution method is not deeply considered. Literature (Mezni & Kbekbi, 2019) uses available process fragments (SPF) as the basic unit for service composition to achieve rapid service composition and improve the reliability of the final solution, a scoring function is proposed to determine the quality level of each SPF and its ability to participate in the combination, finally, this method is experimentally studied and the effectiveness of this method is proved. But this research only uses a single domain feature to solve, and does not use multiple domain features to improve the

solution process. Therefore, this paper proposes a diffraction method based on a priori correlation ABC.

3. SL-GABA ALGORITHM

Users often specify global total constraints for the composition optimization problem to be solved, therefore, in the early stage of the diffraction method, constraint decomposition is required to obtain the corresponding local sub-constraints. Obviously, the constraint decomposition problem is a combinatorial optimization problem. Provide the result of solving the problem (that is, the corresponding constraint information) to the subsequent solving stage of the diffraction method for its use. SLO is a new swarm intelligence optimization algorithm paradigm that simulates the evolution of human intelligence (Liu, Chu, Song et al, 2016). SLO is comprised of three coevolutionary spaces: the bottom layer is the micro space, in this space, individuals in the population undergo corresponding genetic evolution according to certain rules. The middle space is the learning space, individuals in this space learn to increase their intelligence, and provide the more valuable knowledge formed in this space to the upper space, which is the highest space. At the top is the belief space, it transforms the received knowledge in this space into the culture and uses these cultures to influence the genetic evolution of individuals in the bottom layer micro space. (Liu, Chu, Song et al, 2016) has verified through experiments that SLO has good scalability and feasibility as well as its superiority. In addition, social learning theory which is proposed by American psychologist Albert Bandura has already proven that the important effect of learning on human behavior, and Bin Peng proved that cultural influence on the basis of genetic evolution can promote and enhance the results of evolution (Bin, 2005). SLO is a mature optimization algorithm paradigm, we can investigate the specific optimization problem to be solved and can obtain the appropriate optimization algorithm by configuring the SLO accordingly. During this process, the three coevolutionary spaces of SLO need to be configured with corresponding evolutionary algorithms or swarm intelligence algorithms. In this paper, for constraint decomposition, we integrate the improved BA and improved GA algorithm into the learning space and micro-space to obtain a specific SLO algorithm, which is SL-GABA.

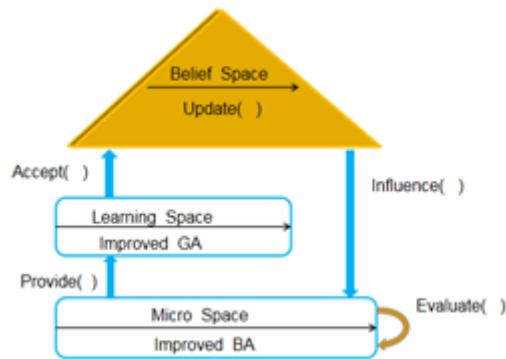
The bat algorithm is a new swarm intelligence random search algorithm that simulates the predation behavior of bats based on their echolocation (Yang, 2010). This algorithm first initializes the initial population in a random manner, then iterates until the stopping condition of the algorithm is met, and performs a local search whenever conditions permit, that is, randomly flying around the optimal individual at the corresponding moment to generate a new individual. The main advantage of the bat algorithm is that it absorbs the advantages of other successful algorithms, parameter adjustment and frequency regulating, on the basis of bat individual echolocation. Literature (Shirjini et al., 2020) analyzes the convergence and stability of this algorithm in the respect of particle dynamics and investigates the dynamics of this algorithm. Literature (Osaba et al., 2016) uses this algorithm to solve the discrete combinatorial optimization problem, which is the traveling salesman problem. Therefore, BA is also suitable as an algorithm for solving the combinatorial optimization problem of constraint decomposition. But it can be seen from the optimization mechanism of this algorithm, in the process of algorithm optimization search, two important operations in this algorithm: individual update and search for local new solutions make the individuals lack appropriate mutation operations, and all individuals are easy to gather near the current optimal individual, causes the individuals to wander in this area, unable to jump out of this area, so the overall global search of this algorithm is limited. So this paper combines this algorithm with other algorithms to ensure the effect of problem-solving.

The genetic algorithm is a search algorithm proposed by simulating the evolution of species in nature and learning from the selection and genetic rules. The global random search of genetic algorithm is its main advantage, and its global random search has self-learning ability. Existing research work has proved that when this algorithm is used to solve combinatorial optimization problems, it is easier to find a better solution in a shorter time. In addition, relevant literature has adopted this algorithm

as an algorithm for solving the combinatorial optimization problem of constraint decomposition and has achieved good results (Liu, Chu, Jia et al, 2016). Therefore, we also use this algorithm as the basis of the algorithm for constraint decomposition.

So, to obtain the corresponding local sub-constraints through constraint decomposition, this paper constructs a micro space based on the improved BA algorithm and a learning space based on the improved GA algorithm under the framework of the SLO algorithm to form an evolutionary algorithm, which is called SL-GABA algorithm. The framework of SL-GABA is shown in Figure 1. The pseudo-code of the SL-GABA algorithm is shown below. The important functions in SL-GABA are defined as follows:

Figure 1. Framework of SL-GABA



Algorithm 1. SL-GABA algorithm

```

// Initialization
Initialize individuals in the bottom micro space according to the improved BA, and evaluate them according to the Evaluate() function
REPEAT
// Individual evolution
Individual evolution is performed based on the improved BA in the bottom micro-space, and provide individuals to the learning space of the middle layer according to the Provide() function
// Individual learning
Perform selection operations, crossover operations, and mutation operations in genetic algorithm
Perform corresponding imitation learning and corresponding observational learning based on improved GA
Extract  $\delta$  eligible individuals from the learning space of the middle layer as valuable knowledge, and these individuals are provided to the highest level of belief space according to the Accept() function, process the original knowledge in the highest level of belief space based on the Update() function
// Cultural influence
When knowledge is accumulated  $\eta$  times and updated culture is obtained, use the updated culture to influence the evolution of each individual in the bottom micro-space based on the influence() function
Record the current best individual
Until the stop condition specified by the user is reached
Evaluate():evaluate the fitness of the individual.
Accept():select  $\delta$  individuals with high fitness values in the middle layer learning space, and send them to the highest level belief space.
Update():use the excellent individuals delivered by the Accept() function to replace the poor individuals in the highest level belief space.
Influence():replace the poor individuals in the lowest level micro-space with the excellent individuals in the highest level belief space.
Provide():provide individuals in the lowest level of micro-space to the middle level of learning space.
    
```

4. DIFFRACTIVE SERVICE COMPOSITION

This section describes a progressive, layer-by-layer service composition method, this method is based on the core service set and the diffraction associated set of each layer. In general, this method consists of three stages: identification of core component services and diffraction-associated component services of each layer, constraint decomposition and service composition based on priori correlation ABC. Each stage will be explained in detail below.

First, mine the associated matrix between each component service according to the correlation set, in this way, the core component services and the diffraction-associated component services of each layer can be found through the associated matrix, and then the process can be divided diffractively. Then, in order to support the diffraction method proposed in this paper, constraint decomposition is required to obtain the corresponding local sub-constraints. At last, based on the sub-constraint of each diffraction combination stage and the corresponding sub-process, use priori correlation ABC to solve until the user's satisfactory solution to the original composition problem is obtained.

4.1. Definitions Of Related Issues

We first define related issues. Explain the formal definition of web service, the formal QoS model of service, and the QoS correlation model between services.

Definition 1 (service):

Each service s is a collection $s=(id_s, G_s, F_s, NF_s)$. The id_s here refers to the id number of s . G_s refers to the basic information of s , such as name information or version information and so on. F_s refers to the function information of s , such as input parameters or output parameters and so on. NF_s refers to non-functional information of s , that is, the qos information of s .

In the real world, web services do not exist in isolation, they often have connections and influences with each other, in other words, QoS information of services is not fixed, it is often affected by other services and fluctuates up and down. So define the formal QoS model of the service.

Definition 2 (correlation-aware QoS model):

QoS is a collection of quality attributes of service s , $QoS_s=\{q_1, q_2, \dots, q_n\}$, and for each q_i , it is a collection:

$$q_i = \left\{ \begin{array}{l} (qd, qa_{s'}, s', cd_{s'}) \\ (qd, qa_{s''}, s'', cd_{s''}), \dots \\ (qd, qa_{s'''}, s''', cd_{s'''}) \end{array} \right\},$$

the qd here refers to the default value of the quality attribute q_i of the service s , $qa_{s'}$ refers to the correlated value of the quality attribute q_i of service s when service s is used together with service s' , s' refers to the service that the value of the quality attribute q_i of service s is correlated to, $cd_{s'}$ refers to the corresponding correlation degree, that is, the frequency of correlation pattern(s, s') appearing in the service history, or the ratio of the number of historical records containing (s, s') to the total number of historical records. If none of s', s'', \dots, s''' is called, then the value of q_i of s is taken as qd , otherwise, take the corresponding correlated value.

Definition 3 (QoS correlation model between services):

The QoS correlation between two services is represented by $(s', s, q_i, cd_{s'})$, that is to say, the value of the quality attribute q_i of the service s is related to whether the service s' is selected and used.

For example, in the service composition process, service s' has been selected and used in advance, then the value of the quality attribute q_i of service s is $qa_{s'}$, otherwise its value is qd .

4.2. identification of core component services and diffraction-associated component services of each layer

Here, first build the associated matrix between each component service, then use the associated matrix to mine the core component services and the diffraction-associated component services of each layer. First, give the formal definition of the associated matrix:

Definition 4 (associated matrix):

The associated matrix is denoted as AM , the rows and columns of the matrix are different component services, $AM = \text{component service} * \text{component service}$, the associated matrix is a symmetric matrix.

$$AM = \begin{bmatrix} AM_{11} & AM_{12} & \dots & AM_{1k} \\ \dots & AM_{2j} & \dots & \dots \\ AM_{1k} & AM_{2k} & \dots & AM_{kk} \end{bmatrix}$$

The associated matrix describes the association relationships between the selection result of each component service, each element AM_{ij} in the matrix represents the strength of the association relationship between the selection result of the component service CS_i and the selection result of CS_j . The algorithm for constructing the associated matrix is given below:

After constructing the associated matrix, the core component services can be identified according to the associated matrix. The method is, calculate $AM_i = \sum_{j=1}^k AM_{ij}$ for each CS_i ($i=1, 2, \dots, k$), when AM_i is greater than a given threshold, CS_i is defined as the core component service. Secondly, determine the associated component services for each diffraction stage, that is, this process is carried out by constructing and running an associated component service mining algorithm.

Algorithm 2. Associated matrix construction algorithm

Input: Correlation set, candidate service sets, threshold χ Output: Associated matrix AM
1. Initialize all elements in AM to 0; 2. For each correlation service in the correlation set do 3. If $correlationdegree \geq \chi$ then 4. Identify the number x of the component service where s is located; 5. Identify the number y of the component service where s' is located; 6. $AM_{xy} = AM_{xy} + \text{correlation degree}$; 7. $AM_{yx} = AM_{yx} + \text{correlation degree}$; 8. End 9. End 10. Return associated matrix AM ;

The algorithm finds and identifies the diffraction-associated component service set of the previous layer layer by layer so that the composition problem-solving algorithm can solve diffractively. When this algorithm gets all component services, this algorithm stops. According to the component service set of each layer, the diffractive solution of the service composition problem can be organized. Lines 2-3 are the initialization of this algorithm, set layer 0 as the core component service set, the variable l refers to the number of the current layer, LCS_l refers to the set of component services identified in layer l , T refers to the total set of component services currently obtained by this algorithm, that is, the union of the set of component services in all layers currently obtained by this algorithm. It is worth noting that, to ensure the connectivity of each diffraction solution sub-process, the service process needs to be scanned to obtain the connected component service set $CNCS_l$ of each layer.

4.3. Decomposition Of Constraint

To support the diffraction method proposed in this paper, it is necessary to obtain the corresponding sub-constraints needed in each diffraction solution stage according to the global total constraints specified by the user for the process of the composition problem. For example, the user gives a global total constraint c_{tot} , it needs to be mapped to k local sub-constraints $\{c_1, \dots, c_k\}$ (k is the number of component services in the process). So the upper bound(lower bound) for the sub-process of each diffraction stage can be obtained according to $\{c_1, \dots, c_k\}$. So we can find that the key question is

Algorithm 3. Associated component services mining algorithm

<p>Input: Associated matrix AM, service process sf, minimum threshold φ, core component service set $CoCS$</p> <p>Output: Associated component service set of each diffraction stage $\{LCS_1, LCS_2, \dots\}$</p>
<ol style="list-style-type: none"> 1. Begin 2. $T \leftarrow \emptyset, l \leftarrow 0, LCS_l \leftarrow CoCS$ 3. $T \leftarrow T \cup LCS_l$ 4. While $\exists CS_i \in sf.CS \cap CS_i \notin T$ do//$sf.CS$ refers to the collection of all component services in the service process sf 5. $l \leftarrow l + 1$ 6. $LCS_l \leftarrow \emptyset$ 7. For each component service $CS_j \in LCS_{l-1}$ in LCS_{l-1} do 8. Read $AM_{r_j} (r = 1, \dots, k)$ in the associated matrix AM, and identify matrix elements greater than the minimum threshold to determine the associated component service set ACS_j corresponding to CS_j 9. $LCS_l \leftarrow LCS_l \cup ACS_j$ 10. End 11. Scan the service process, determine the connected component service set $CNCS_l$ of LCS_l 12. $LCS_l \leftarrow LCS_l \cup CNCS_l$ 13. $LCS_l \leftarrow LCS_l - T$ 14. $T \leftarrow T \cup LCS_l$ 15. If ($LCS_l = \emptyset$) 16. Break 17. End 18. Return associated component service set $\{LCS_1, LCS_2, \dots\}$ 19. End

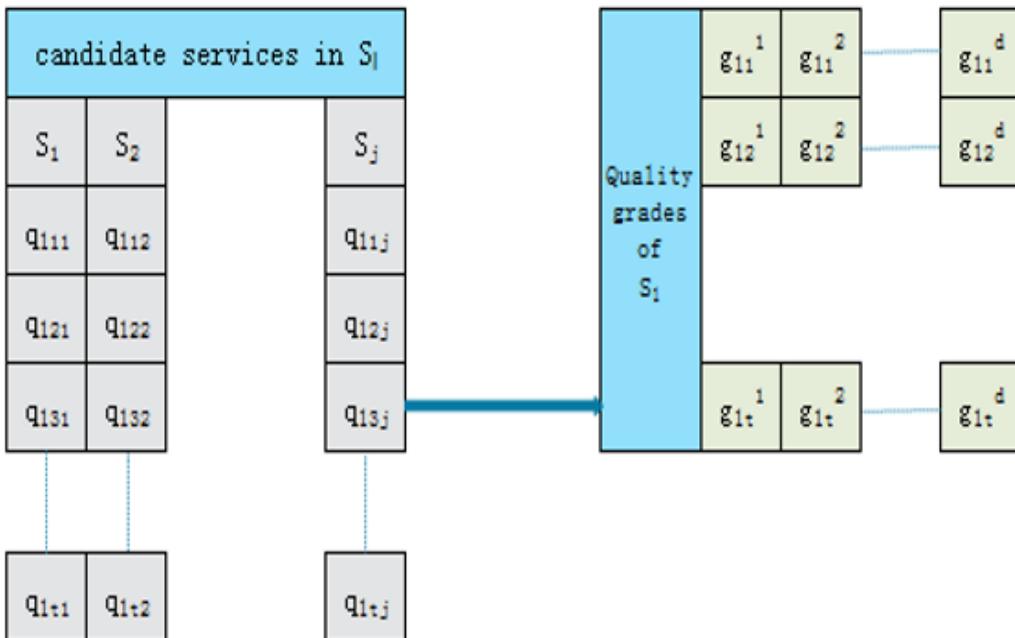
how to obtain $\{c_1, \dots, c_k\}$. It should be noted that it needs to satisfy the following requirements in the process of constraint decomposition: (1) While satisfying all local sub-constraints, it also satisfies the global total constraint. (2) All local sub-constraints should be as relaxed as possible, therefore, as many concrete services as possible can satisfy these local sub-constraints so as to prevent the omission of some concrete services that may become part of the optimal solution. For effective constraint decomposition, the value domain of each attribute in each service class needs to be divided into discrete a series of qos values, we call these discrete qos values quality grades. In this way, each global total constraint can be mapped to a series of quality grades, and we use these quality grades as local sub-constraints. Generally, more than one (such as L) global total constraints are proposed by a user. So each component service should satisfy L local sub-constraints. The purpose of the constraint decomposition problem in this paper is to determine the best quality grade combination for each component service. So first define the corresponding evaluation function for the quality grade combination.

4.3.1. Evaluation Function for Quality Grade Combination And Objective Function For Constraint Decomposition

As shown in Figure 2, we divide the value domain of each attribute q_i in service class S_i into discrete a series of qos values, so the quality grades of S_i are obtained. q_{ij} refers to the t -th quality attribute value of the j -th concrete service in service class S_i , g_t^d refers to the d -th quality grade of the t -th quality attribute in S_i .

Because the concrete services in each service class contain more than one quality attributes, and each quality attribute corresponds to a series of quality grades, therefore, these quality grades have formed many quality grade combinations. To compare these quality grade combinations, an evaluation function $F(QGC_{im})$ is proposed to evaluate the fitness value of quality grade combinations. $F(QGC_{im})$ is defined as follows:

Figure 2. The process of producing quality grades



$$F(QGC_{lm}) = \frac{n(QGC_{lm})}{n_{tot}} \times \frac{U_{avg}(QGC_{lm})}{U_{max}}$$

QGC_{lm} here refers to the m -th quality grade combination of service class S_l . $n(QGC_{lm})$ refers to the total number of concrete candidate services that meet this quality grade combination, n_{tot} refers to the number of all concrete candidate services in S_l . $U_{avg}(QGC_{lm})$ refers to the average value of the utility of all concrete candidate services under the constraint of this quality grade combination, U_{max} refers to the maximum value of the utility of all concrete candidate services in S_l .

As mentioned before, the purpose of constraint decomposition in this paper is to determine the best quality grade combination for each component service. So the objective function of the constraint decomposition problem in this paper is:

$$Maximize \left(\sum_{l=1}^k F(QGC_{lm}) \right), 1 \leq m \leq d$$

It can be seen that the constraint decomposition problem in this paper is a combinatorial optimization problem, therefore, we design the SL-GABA algorithm based on the SLO algorithm paradigm and use this algorithm for constraint decomposition. In the following, the relevant parts and elements of the SL-GABA algorithm for constraint decomposition will be explained in detail.

4.3.2. Coding of SL-GABA

The purpose of the constraint decomposition problem in this paper is to determine a qualified and best quality grade combination for each component service. Because each component service needs to be assigned a quality grade combination, and a process of composition problem often has more than one component services, so use the combination of quality grade combination of each component service to express the solution of this problem. So adopt a two-dimensional coding for the SL-GABA algorithm, this coding is shown in Figure 3, $\{CS_1, CS_2, \dots, CS_k\}$ here are component services, $\{Q_1, Q_2, \dots, Q_L\}$ are the quality attributes involved in the constraints to be decomposed. $Q_{kL}G_r$ is the r -th quality grade of the L -th quality attribute of the component service CS_k . Any row in this figure represents the quality grade combination of its corresponding component service.

4.3.3. Initialize the Population for SL-GABA

First generate a new individual in the SL-GABA algorithm: select any quality grade of each quality attribute of a component service, thus forming a quality grade combination for this component service, then combine the quality grade combination of CS_1, CS_2, \dots, CS_k to generate an individual. After getting this individual, this individual needs to be examined to see if it is reasonable and valid, if not, regenerate an individual.

Repeat the above process until n_c reasonable valid individuals are produced, n_c here is the number of individuals competing each time. Evaluate the quality of these n_c individuals according to fitness, put the best individual into the initial population. Iterate this process until the inserted individuals fill up the initial population.

Here, it is reasonable and valid when an individual satisfies the following two points at the same time: (a) Regardless of which row of this individual, you can find concrete services that do not violate any local sub-constraint in the quality grade combination corresponding to this row. (b) Regardless of which column of this individual, the aggregation of this column does not violate the global total constraints corresponding to this column.

Figure 3. The coding of SL-GABA

	Q_1	Q_2	Q_3	...	Q_L
CS_1	$Q_{11}G_i$	$Q_{12}G_p$	$Q_{13}G_x$...	$Q_{1L}G_f$
CS_2	$Q_{21}G_j$	$Q_{22}G_q$	$Q_{23}G_y$...	$Q_{2L}G_z$
CS_3	$Q_{31}G_1$	$Q_{32}G_t$	$Q_{33}G_z$...	$Q_{3L}G_h$
\vdots	\vdots	\vdots	\vdots	...	\vdots
CS_k	$Q_{k1}G_r$	$Q_{k2}G_u$	$Q_{k3}G_v$...	$Q_{kL}G_r$

4.3.4. Individual Evolution of Micro Space

For the bat algorithm in micro space, the position is the expression of the solution of the problem to be solved, and the change of individual position is related to the real-time individual speed. For the speed update in the original bat algorithm, the speed of the current generation is related to the speed of the previous generation, therefore, if the speed of the previous generation leads the individuals of the current generation to deviate from the right direction, bring interference to the evolution of individuals.

Therefore, two update methods are used for the speed update. The first update method uses the update method in the original algorithm, the second update method uses a new update method.

In the first update method, update the speed of bat individual according to the rule defined in (Yang, 2010), the speed of the t generation is updated as follows:

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i$$

Here x_* is the current global optimal solution obtained after comparison.

In the second update method, to ensure that the search directions of the bat individuals are toward the current global optimal solution and avoid the direction deviation that may be caused by the speed of the corresponding previous generation, the operation of this update method is as follows:

$$v_i^t = (x_i^t - x_*)f_i$$

So in this paper, individuals of $m/2$ in the population use the first update method to update, the remaining $m/2$ individuals are updated using the second update method. So the speed update method of the entire population in this paper is as follows:

$$v_i^t = \begin{cases} v_i^{t-1} + (x_i^t - x_*)f_i & \text{if } (i \leq m / 2) \\ (x_i^t - x_*)f_i & \text{otherwise} \end{cases}$$

4.3.5. individual learning of learning space

The imitation learning and observational learning in the learning space are as follows:

(1) imitation learning

For each individual in the current population:

- a. Select a fixed number(n_1) of outstanding individuals from the current population according to the tournament method, put these individuals together with the current individual to get a mutual learning individual group.
- b. Divide all individuals in the mutual learning individual group into the same number of segments.
- c. For each segment of all individuals in the mutual learning individual group, compare them to find the best segment and use the best segment to construct an individual.
- d. Compare the newly constructed individual with the current individual, retain the better individual.

The imitation learning in this paper is shown in Figure 4, I_i here is the current individual, I_1, I_2, \dots, I_{n_1} are n_1 individuals in the mutual learning individual group, the colored parts are the best ones in the corresponding segments, I' is an individual constructed using the best segments, at last, the better individual I_F is retained by comparison using step d.

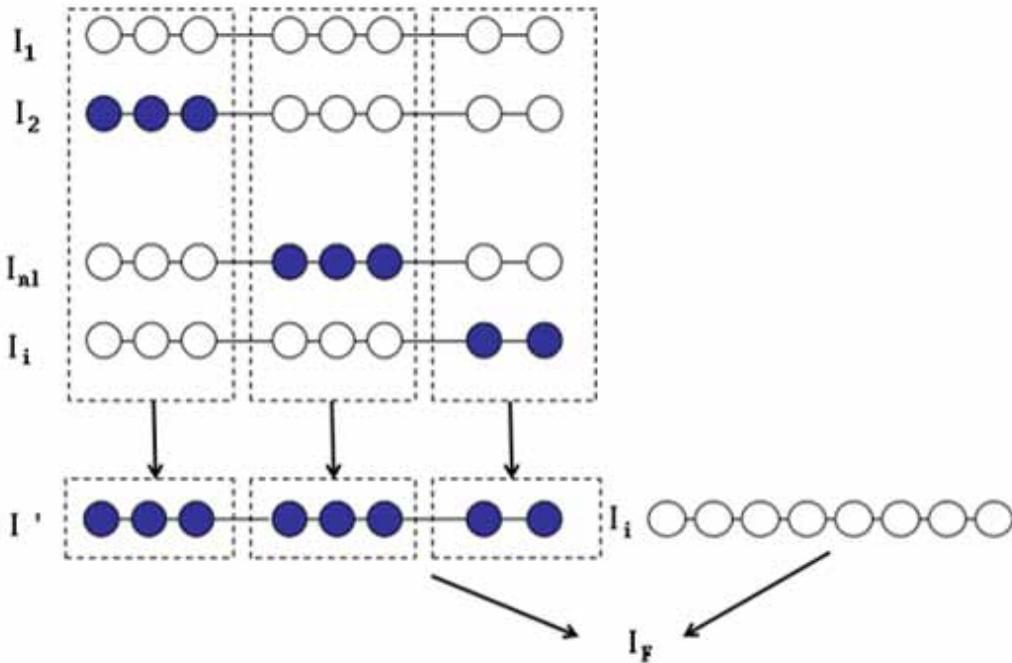
(2) observational learning

As we know, human learning is a process of acquiring knowledge and skills in social life and practice in the active interaction with others who are better than oneself, so as to greatly improve the ability of learning individuals. So the guiding ideology of observational learning in this paper is that every individual in the current group learns from individuals who are better than themselves, specific steps are as follows:

For each individual in the current population:

- a. Put the individuals who are better than the current individual in the population with the current individual to get a mutual learning individual group.
- b. Divide all individuals in the mutual learning individual group into the same number of segments.
- c. For each segment of all individuals in the mutual learning individual group, compare them to find the best segment and use the best segment to construct an individual.
- d. Compare the newly constructed individual with the current individual, retain the better individual.

Figure 4. Imitation learning



4.3.6. Constraint Decomposition Based on SL-GABA

After building SL-GABA, use this algorithm for constraint decomposition. The pseudo code of the SL-GABA-based constraint decomposition algorithm is as follows:

4.4. Priori Correlation ABC Algorithm

In our previous research work, taking into account the advantages of the ABC algorithm and the role of the features in the service domain in solving optimization problems in the domain, select ABC as the basic algorithm of the model, deeply analyze the operation principle of this algorithm and the specific influence of the features in the service domain on the problem solving, and improve the main optimization strategy of this algorithm, thus propose the S-ABC paradigm(Xu et al., 2017). When solving optimization problems in the domain, this paradigm can provide an effective algorithm model and specific optimization strategies. So this paper generates an applicable algorithm based on this paradigm and the influence of the features in the service domain on the solution of the sub-problems involved in the diffraction solution process, which is the priori correlation ABC algorithm.

We have pointed out in the literature(Xu et al., 2017) that in the process of continuous development and evolution of service applications, the domains involved in each service industry have gradually formed and displayed unique domain features. For the diffractive service composition method proposed in this paper, consider the relationship between core component services and diffraction-associated component services, therefore the domain feature of correlation is involved in the various stages of the diffraction solution. In addition, with the continuous operation of the service system, relevant historical data can be selected from a large number of historical operation records, then get valuable service usage experience. Therefore, in each stage of the diffraction solution, it also involves the domain feature of priori. In the sub-processes corresponding to the sub-problems involved in the diffraction solution process, each specific candidate service in the candidate service set corresponding to each

Algorithm 4. Constraint decomposition based on SL-GABA

<p>Input: Number of component services k, the qos of the services corresponding to each component service, global constraints $C = \{C_1, \dots, C_i, \dots, C_L\}$, number of individuals in the population n_{pop}, the number of outstanding individuals to be selected in order to construct a mutual learning individual group n_l, the maximum number of iterations N_{max}, number of competing individuals n_c, the various parameters of the bat algorithm, various parameters of genetic algorithm, the number of individuals extracted from the learning space δ, the knowledge accumulation threshold η of the highest level belief space Output: Local sub-constraints for each component service</p>
<pre> // Initialization // Initialize n_{pop} individuals in the bottom micro space according to the improved BA; For $i=1$ to n_{pop} Do Initialize an individual in the bottom micro space according to 4.3.3; Evaluate this individual according to the Evaluate() function; EndFor // Individual evolution For $i=1$ to $n_{pop}/2$ Do Perform individual evolution in the bottom micro-space according to the first speed update method in 4.3.4; EndFor For $i=1$ to $n_{pop}/2$ Do Perform individual evolution in the bottom micro-space according to the second speed update method in 4.3.4; EndFor Provide individuals to the learning space of the middle layer according to the Provide() function; // Individual learning Perform selection operations, crossover operations, and mutation operations in the genetic algorithm; For $i=1$ to n_{pop} Do Perform corresponding imitation learning based on the method in 4.3.5; EndFor For $i=1$ to n_{pop} Do Perform corresponding observational learning based on the method in 4.3.5; EndFor Extract δ qualified individuals from the learning space of the middle layer as valuable knowledge; Provide these individuals to the highest layer belief space according to the Accept() function; Process the original knowledge in the highest level belief space based on the Update() function; // Cultural influence If(knowledge accumulation times= η) Then Execute the Influence() function; EndIf Record the current best individual; If(number of iterations= N_{max}) Then Output the current best individual; EndIf Else number of iterations= number of iterations+1; Go to the evolutionary part of the individual; EndIf </pre>

component service has the same function and similar QoS, therefore, in each stage of the diffraction solution, it also involves the domain feature of similarity.

Therefore, under the premise of following the paradigm framework and specific optimization strategies in the literature (Xu et al., 2017), make full use of the influence of these three domain features in the solution of each sub-problem to generate a priori correlation ABC algorithm. First, an algorithm that considers priori and similarity is generated based on this paradigm, which is the priori ABC algorithm.

Secondly, considering the influence of correlation in the solution of each sub-problem, the priori correlation ABC algorithm is proposed based on the priori ABC algorithm. Consider the influence of correlation on problem solving, we need to carry out corresponding detailed design for the initial food source generation part and the scout bee part of the algorithm generated above, regardless of whether the algorithm is in the search stage of the priori service scheme set and priori service set, or the search stage of the similar service set, or the search stage of the general service set, that is, the step1, step4, step7, step10, step13, and step16 parts in the algorithm. Since the task of the initial food source generation part and the scout bee part is to ensure the diversity of food sources and avoid falling into local optimality, therefore, the detailed design of these two parts only considers the mandatory correlation of SC and does not consider the correlation of QC, in other words, a certain amount of QoS sacrifice in exchange for the diversity of the entire bee colony. The SC type correlation here refers to the correlation between the selection results of different component services in the composition process, QC type correlation means that the quality value of the service depends on the selection result of other component services.

For the initial food source generation part and the scout bee part, when generating an initial food source(new food source), the SC type correlation relationship between services should be considered, if a candidate service $s_{p,j}$ has been selected, and its SC type correlation service set exists, then the candidate service selected for CS_{p+1} should be limited to this correlation service set. So the detailed correlation-aware single food source initialization(update) algorithm is as follows:

It is worth noting that, when the algorithm is in the search stage of priori service scheme set and priori service set, the corresponding candidate service set here refers to the priori service set, when the algorithm is in the search stage of the similar service set, it refers to the similar service set, when the algorithm is in the search stage of general service set, it refers to the general service set.

Consider the influence of correlation on problem solving, we also need to modify the employed bee part and the onlooker bee part of the algorithm accordingly, regardless of whether the algorithm is in the search stage of priori service scheme set and priori service set, or the search stage of the similar service set, or the search stage of the general service set, that is, the step2, step3, step8, step9, step14, and step15 parts in the algorithm. So for the employed bee part and the onlooker bee part, once through the neighborhood search, it is decided to use the generated new food source v_i to replace the original food source x_i , that is to say, replace the p-th dimension $s_{p,j}$ of x_i with $s'_{p,j}$ to generate v_i , then modify the employed bee phase and the onlooker bee phase in the algorithm to produce the following algorithm(correlation-aware employed bee phase/onlooker bee phase algorithm):

In this algorithm, first check the correlation of the SC type, then check the correlation of QC type, that is to say, the two have different priorities. This is because, once a certain SC type correlation works, then the related QC type correlations will be invalid. If the SC type correlation service set of x_i exists, then select the candidate service that can optimize the food source from this correlation service set, and use the food source formed by this candidate service to replace x_i . If the SC type correlation service set of x_i does not exist, but its QC type correlation service set exists, then select the candidate service that can optimize the food source from this correlation service set, compare the food source formed by this candidate service with x_i , and keep the better one. So far, we have generated a priori correlation ABC algorithm.

5 EXPERIMENTS

This part uses experiments to verify the superiority and influencing factors of the algorithm proposed in this paper.

Algorithm 5. Priori ABC algorithm

<p>Input: Priori service scheme set, priori service sets, similar service sets, general service sets, α, β, SN, search step, limit, arbitration criteria, transition condition Output: The current optimal solution that satisfies the arbitration criteria</p>
<pre> //Priori service scheme set and priori service set search stage Step 1:Generate $\alpha \times SN$ and $\beta \times SN$ initial food sources based on priori service scheme set and priori service sets respectively; Do{ Step 2: For each food source x_i {Generate a new food source v_i according to the search direction and search step in the prior service set; If (fitness(v_i)>fitness(x_i)) $x_i = v_i$; $et_i, et_i = 0$; Else $et_i, et_i = et_i + 1$; } Step 3: Calculate the probability for each food source x_i; For each onlooker bee b_j {Select a food source x_i according to the probability for further exploration; This onlooker bee b_j is converted to an employed bee to mine the food source x_i; } Step 4: If limit=et_i for a food source x_i {Generate a new food source x_i' based on priori service sets and method FS(PriS); $x_i = x_i'$; $et_i, x_i = 0$; } Step 5: Memorize the best food source achieved so far; Step 6: If the arbitration criterion is satisfied Return the best food source achieved; } while the transition condition is not met //Similar service set search stage Step 7: Generate SN initial food sources based on similar service sets; Do{ Step 8: For each food source x_i {Generate a new food source v_i according to thesearch direction and search step in the similar service set; If (fitness(v_i)>fitness(x_i)) $x_i = v_i$; $et_i, et_i = 0$; Else $et_i, et_i = et_i + 1$; } Step 9: Calculate the probability for each food source x_i; For each onlooker bee b_j {Select a food source x_i according to the probability for further exploration; This onlooker bee b_j is converted to an employed bee to mine the food source x_i; } Step10: If limit=et_i for a food source x_i {Generate a new food source x_i' based on similar service sets and method FS(SimSs); $x_i = x_i'$; $et_i, x_i = 0$; } Step11:Memorize the best food source achieved so far; Step12:If the arbitration criterion is satisfied Return the best food source achieved; } while the transition condition is not met //General service set search stage Step13:Generate SN initial food sources based on general service sets; Do{ Step14:For each food source x_i {Generate a new food source v_i in a random manner in the general service set; If (fitness(v_i)>fitness(x_i)) $x_i = v_i$; $et_i, et_i = 0$; Else $et_i, et_i = et_i + 1$; } Step15:Calculate the probability for each food source x_i; For each onlooker bee b_j {Select a food source x_i according to the probability for further exploration; This onlooker bee b_j is converted to an employed bee to mine the food source x_i; } Step16: If limit=et_i for a food source x_i {Generate a new food source x_i' based on general service sets and method FS(GenSs); $x_i = x_i'$; $et_i = 0$; } Step17:Memorize the best food source achieved so far; Step18:If the arbitration criterion is satisfied Return the best food source achieved; } while the transition condition is not met </pre>

5.1 Experimental Setting

In this experiment, our experimental subject is a common travel service composition problem in daily

Algorithm 6. Correlation-aware single food source initialization(update) algorithm

Input: Candidate service sets, SC type correlation service set, the number of component services in the process k Output: Updated (initialized) food source
<pre> 1. var binSet={CS₁, CS₂,...,CS_k}; 2. int p=0; 3. Select randomly s_{1,j} from the corresponding candidate service set of CS₁, and bind s_{1,j} to CS₁, binSet = binSet -{CS₁}; 4. p=p+1; 5. While (binSet!=empty) 6. { If s_{p,j} has a SC correlation set 7. Select randomly s_{p+1,j} from this correlation set, and bind s_{p+1,j} to CS_{p+1}, binSet = binSet -{CS_{p+1}}, p=p+1; 8. Else 9. Select randomly s_{p+1,j} from the corresponding candidate service set of CS_{p+1}, and bind s_{p+1,j} to CS_{p+1}, binSet = binSet -{CS_{p+1}}, p=p+1; } </pre>

life, the component services in the process corresponding to this problem are insurance component service, long-distance transport component service, short-distance transport component service, hotel component service, catering component service, and scenic spot component service. Although the structure of this travel service composition problem is sequential, considering that any structure can be transformed into a sequential structure(Ardagna & Pernici, 2007) (Jang et al., 2006), therefore, the choice of travel service composition problem as the experimental subject is universally representative. The insurance component service corresponds to 57 candidate web services, the long-distance transport component service corresponds to 51 candidate web services, the short-distance transport component service corresponds to 58 candidate web services, hotel component service corresponds to 60 candidate web services, catering component service corresponds to 64 candidate web services, scenic spot component service corresponds to 71 candidate web services. Some of the QoS attribute values of these candidate web services are real attribute data collected from Ctrip.com(<https://www.ctrip.com/>) and Dianping.com(<http://www.dianping.com/>), other QoS attribute values are attribute data obtained through simulation. Each candidate web service corresponds to four QoS attributes, that is, price, reliability, reputation and time. Assume that the user's preferences for price, reliability, reputation and time are 0.4, 0.2, 0.1, 0.3. The global constraint specified by the user is: price ≤ 4000. All algorithms in the experimental part of this paper are implemented by Python, the configuration of the computer is Windows 7 operating system, Intel(R) Core(TM) i5-4200U CPU@1.60GHz 2.30GHz, 4.00GB RAM.

5.2 Performance Comparison

Based on the experimental setting described above, we compare the algorithm proposed in this paper, RGA(Wu et al., 2014) and TBA(Wang et al., 2013). In order to avoid the variability of various factors in the algorithm operating environment leading to the variability of algorithm performance, and ensure the objectivity of performance comparison experiments, for each comparison algorithm, the average of the results of multiple algorithm running is used as the final data for the performance comparison of this algorithm. In order to make the comparison charts more intuitive, expand the vertical axis evaluation values in the comparison charts by 1000 times.

Algorithm 7. Correlation-aware employed bee phase/onlooker bee phase algorithm

<p>Input: Candidate service sets, search step, SC type correlation service set, QC type correlation service set Output: Updated food source</p>
<ol style="list-style-type: none"> 1. For each food source x_i 2. {Generate a new food source v_i according to the corresponding operation in the corresponding candidate service set; 3. If (fitness(x_i)>fitness(x_i)) 4. { x_i x_i ; 5. If x_i has a SC correlation set 6. { For (h=1 to RS. size)//SC correlation set's size 7. {Select $s_{p+1,h}$ from SC correlation set , and bind $s_{p+1,h}$ to CS_{p+1} to form a new food source v_{ih}' ; 8. Evaluate the fitness of v_{ih}' ; } 9. Memorize the best food source achieved so far; 10. x_i x_i the best food source; } 11. Else 12. { If x_i has a QC correlation set 13. { max=0; 14. For (h=1 to RS. size) //QC correlation set's size 15. { Select $s_{p+1,h}$ from QC correlation set, and bind $s_{p+1,h}$ to CS_{p+1} to form a new food source v_{ih}' ; 16. Evaluate the fitness of v_{ih}' ; } 17. For(h=1 to RS. size) //QC correlation set's size 18. If (fitness(v_{ih}')>max) 19. max=fitness(v_{ih}'); 20. If(max>fitness(x_i)) 21. x_i x_i v_{ih}' ; } 22. x_i x_i 0; } 23. Else 24. x_i x_i +1; }

The parameter configuration of the SL-GABA algorithm in the method proposed in this paper is: $\delta =5$, $\eta =5$, $nc=5$, $nl=5$, $npop=10$, $N_{max}=500$, $\alpha =0.9$, $\gamma =0.9$, the crossover probability is 0.8, the mutation probability is 0.5. The parameter configuration of the priori correlation ABC algorithm in the method proposed in this paper is: $\alpha =0.2$, $\beta =0.8$, $SN=20$, $limit=15$, $transition\ condition=0.007$. The parameter configuration of the RGA is: $popsize=100$, the crossover probability is 0.7, the mutation probability is 0.3. The parameter configuration of the TBA is: its parameter

configuration is the same as the parameter configuration of the priori correlation ABC in this paper. In addition, because it is necessary to prepare the priori service scheme set, priori service sets, similar service sets, and general service sets for the priori correlation ABC algorithm, so first we can iterate ABC 1000 times to generate 1000 feasible solutions. Secondly, according to the standard definition of the priori service scheme, the priori service scheme set is obtained. At last, according to the standard definition of priori service set, similar service set and general service set, get three service sets for each component service in the service process. This section uses three arbitration criteria to compare the performance of the three algorithms.

(1) maximum running time arbitration criteria

When the running time of the solving algorithm reaches the maximum running time specified by the user, return the current global optimal solution. Figure 5 shows the experimental results. The vertical axis in this figure is the evaluation value of the solution returned to the user by each algorithm in the corresponding running time, the horizontal axis is the maximum running time specified by the user.

We can see that, under the same maximum running time, the number of times the solutions returned to the user by the algorithm proposed in this paper better than the solutions returned to the user by other algorithms is 20 times. Therefore, it can be concluded that, under the maximum running time arbitration criteria, the performance of the algorithm proposed in this paper is better than other algorithms.

(2) given repetition numbers arbitration criteria

When the current global optimal solution obtained by the solving algorithm satisfies the given repetition numbers, return the current global optimal solution. Figure 6 shows the experimental results. The vertical axis in this figure is the evaluation value of the solution returned to the user by each algorithm under the corresponding repetition numbers, the horizontal axis is the repetition numbers specified by the user.

We can see that, under the same repetition numbers, the number of times the solutions returned to the user by the algorithm proposed in this paper better than the solutions returned to the user by other algorithms is 20 times. Therefore, it can be concluded that, under the given repetition numbers arbitration criteria, the performance of the algorithm proposed in this paper is better than other algorithms.

Figure 5. Maximum running time arbitration criteria

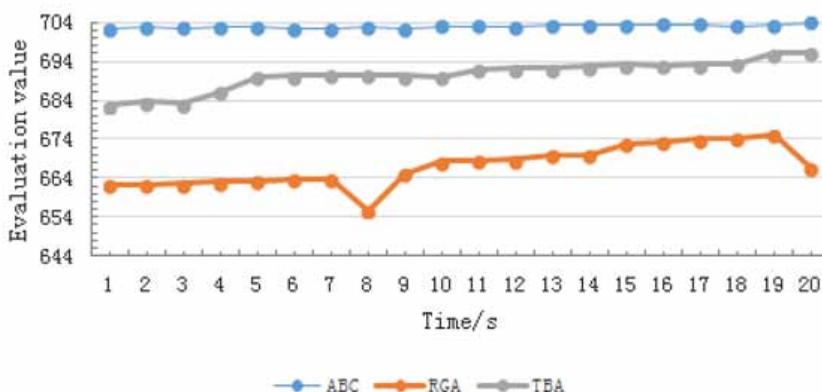
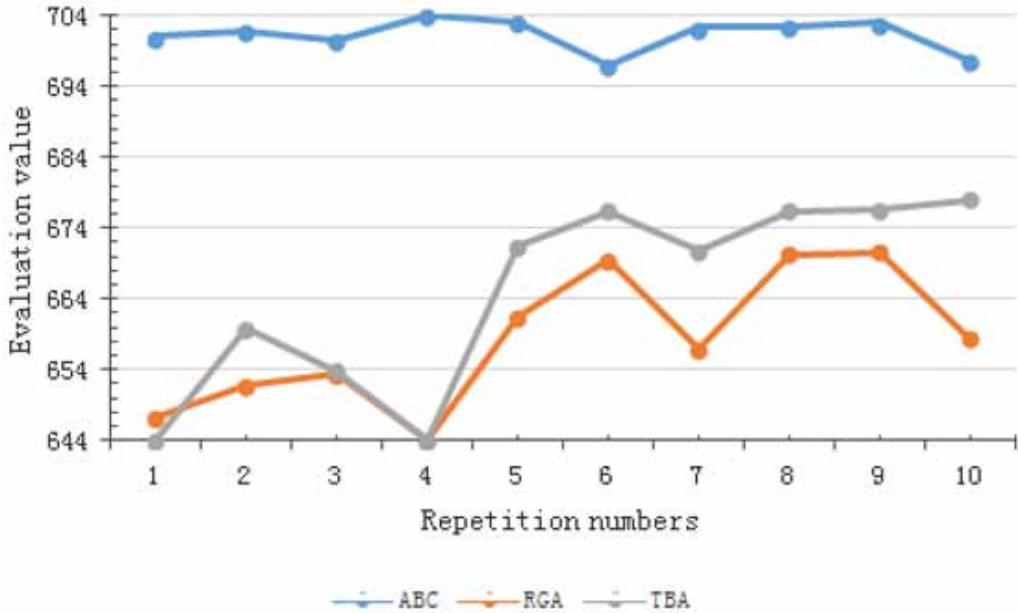


Figure 6. Given repetition numbers of arbitration criteria



(3) given iteration numbers arbitration criteria

When the iteration numbers of the solution algorithm meets the given iteration numbers, return the current global optimal solution. Figure 7 shows the experimental results. The vertical axis in this figure is the evaluation value of the solution returned to the user by each algorithm under the corresponding iteration numbers, the horizontal axis is the iteration numbers specified by the user.

We can see that, under the same iteration numbers, the number of times the solutions returned to the user by the algorithm proposed in this paper better than the solutions returned to the user by other algorithms is 20 times. Therefore, it can be concluded that, under the given iteration numbers arbitration criteria, the performance of the algorithm proposed in this paper is better than other algorithms.

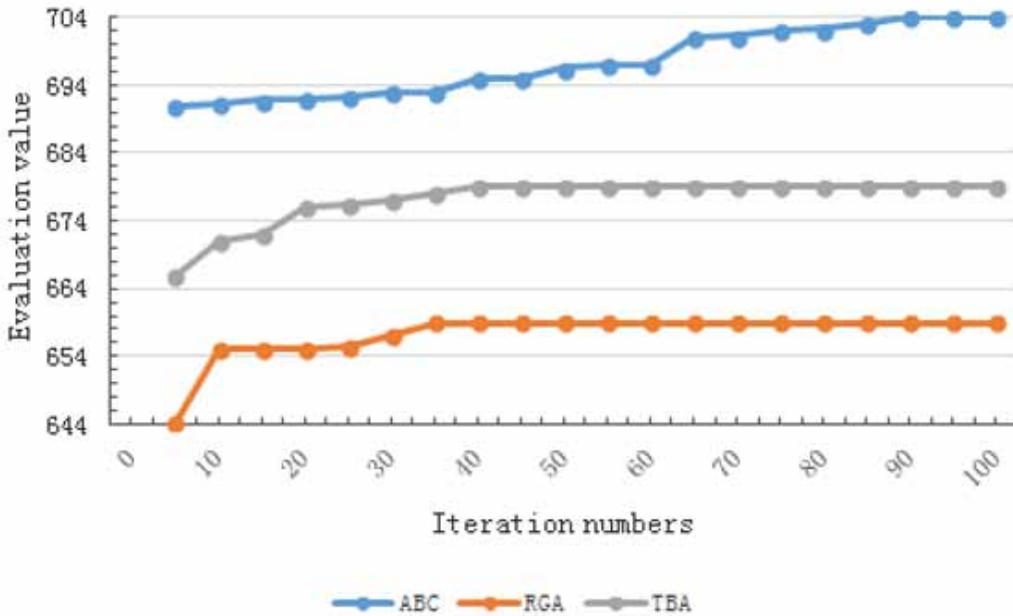
It can be found from the above comparative experiments, no matter which arbitration criteria is used, the algorithm proposed in this paper is better than the other two algorithms in performance. Compare with these two algorithms, the algorithm proposed in this paper can obtain an approximate optimal solution in less time.

5.3 Influencing Factor Verification

The main influencing factors of the algorithm proposed in this paper include related characteristics of correlation set, priori service set and similar service set, here their relevant characteristics are: the size of the correlation set RSS, the correlation degree of correlation set RSCD, the size of the priori service set PS, the credibility of priori service set PC, the usage frequency of priori service set PF, the size of similar service set SSI. Their calculation formulas are:

(1) $RSS = \frac{1}{k} \times \sum_{i=1}^k RSS_i$, RSS_i represents the sum of the size of the correlation set of candidate services in the i-th component service.

Figure 7. Given iteration numbers of arbitration criteria



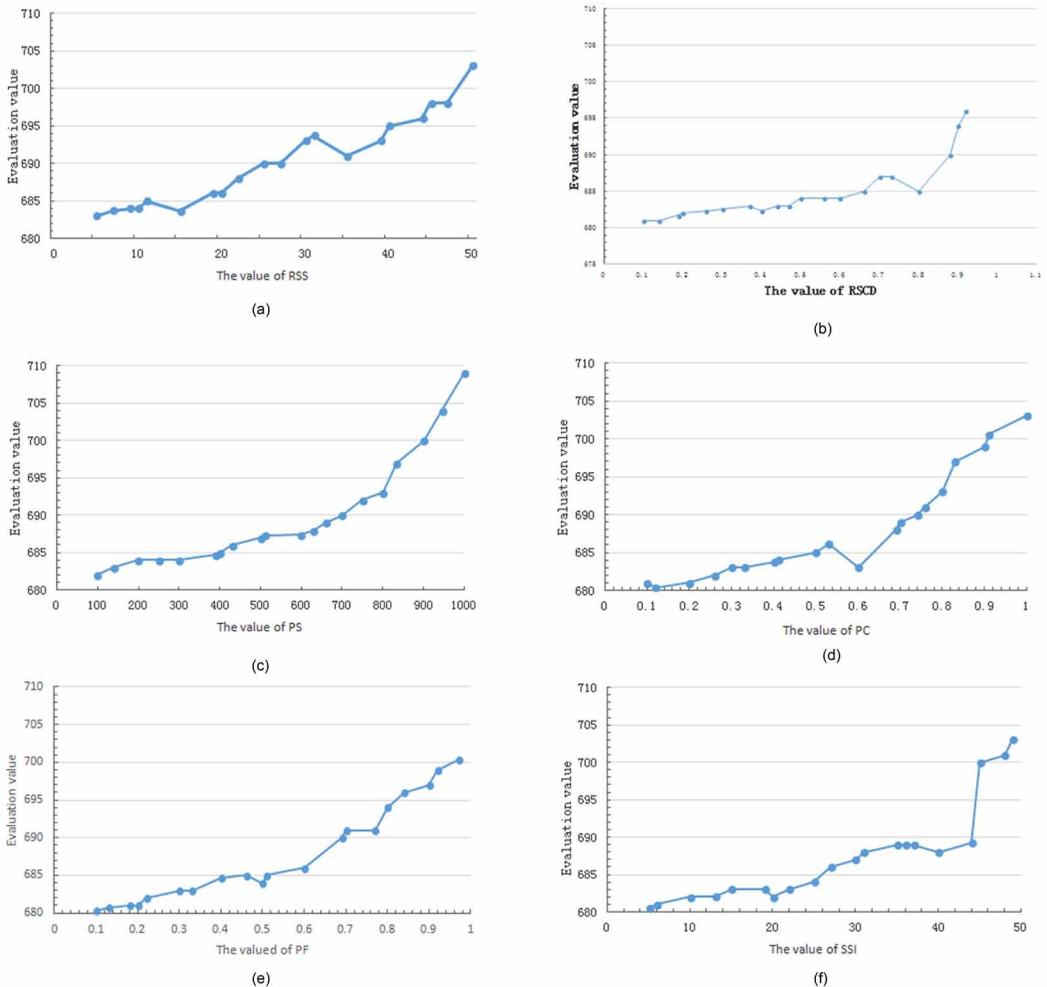
- (2) $RSCD = \frac{1}{k} \times \sum_{i=1}^k RSCD_i$, $RSCD_i$ represents the correlation degree of the correlation set of the i -th component service, $RSCD_i = \frac{1}{RSS_i} \times \sum_{j=1}^{RSS_i} cd_j$, cd_j represents the correlation degree of the j -th correlation service in the correlation set corresponding to the i -th component service.
- (3) $PS = \frac{1}{k} \times \sum_{i=1}^k PS_i$, PS_i represents the size of the priori service set of the i -th component service.
- (4) $PC = \frac{Ite}{MaxIte}$, Ite represents the number of iterations of the algorithm used to generate this set, $MaxIte$ represents the maximum number of iterations of the algorithm that generates this set.
- (5) $PF = \frac{1}{k} \times \sum_{i=1}^k PF_i$, $PF_i = \frac{\left(\sum_{j=1}^{PS_i} UN_j \right)}{\left(PS_i \times HDS \right)}$, UN_j represents the number of times the j -th priori service in the priori service set of the i -th component service has been called in the historical records, HDS indicates the number of historical records.
- (6) $SSI = \frac{1}{k} \times \sum_{i=1}^k SSI_i$, SSI_i represents the size of the similar service set of the i -th component service, $SSI_i = \left| \bigcup_{j=1}^{PS_i} Sim(ps_j) \right|$, ps_j represents the j -th priori service in the priori service set of i -th component service, $Sim()$ is a function to obtain the similar service set of ps_j , $| \cdot |$ is a function to get the size of a service set.

Use ABC to solve the experimental subject described in Section 5.1, randomly get different historical information according to different iteration times. In this paper, first generate 100 different historical information, then randomly extract 20 different historical information from it for experimentation, and number them respectively as HI1, HI2, ..., HI20, and calculate their eigenvalues. Use the 20 extracted historical information to obtain the corresponding priori service sets, correlation sets, and similar service sets, these sets can be used directly when using the algorithm proposed in this paper to solve the problem.

The following experiments are used to verify the influence of eigenvalues PS, PC, PF, RSS, RSCD, SSI on the performance of the algorithm, Figure 8 shows the experimental results. The horizontal axis is different eigenvalues, the vertical axis is the evaluation values of the solutions. The parameter configuration of the algorithm is the same as the parameter configuration in Section 5.2. In addition, same as Section 5.2, the average of the results of multiple algorithm running is used as the final data for the performance comparison of this algorithm. In order to make the comparison charts more intuitive, expand the vertical axis evaluation values in the comparison charts by 1000 times.

We can see from the figure above, with the increase of PS, PC, PF, RSS, RSCD, SSI, the algorithm proposed in this paper has an enhanced ability to search for the optimal solution. The

Figure 8. (a – f) The influence of eigenvalues on the performance of the algorithm



main reasons are divided into two aspects: first, this paper adopts the solution strategy of diffractive service composition method, therefore, the higher the value of RSS and RSCD, the more accurate the identification of core component services and the diffraction-associated component services of each layer, the greater the useable value, this makes the diffractive division of the process more reliable, so this makes the algorithm's ability to search for optimal solution stronger. Secondly, this paper uses priori correlation ABC algorithm as the solution algorithm, therefore, the greater the value of PS, PC, PF, RSS, RSCD, SSI, the algorithm makes fuller and more effective use of priori, correlation, and similarity, so this makes the algorithm's solution better. In short, the experimental results show that when the priori service set is richer, the value of credibility and usage frequency are larger, the size of the similar service set is larger, the size of the correlation set is larger, and the correlation degree is larger, the better the performance of the algorithm proposed in this paper.

CONCLUSION AND FUTURE WORK

This paper proposes a diffraction method based on priori correlation ABC, the work of this paper is divided into the following three aspects: (1) A diffractive service composition method is proposed to solve the problem of service composition in a diffractive manner, in this way, the effect and efficiency of problem solving are improved by considering the impact of core component services and diffraction-associated component services of each layer on the algorithm solving process. (2) Taking into account the advantages of ABC in all aspects, this algorithm is selected as the basic algorithm for problem solving, deeply dig into the influence of multiple domain features on all aspects of the algorithm solving, improve ABC's related optimization strategies and optimization operations, a priori correlation ABC algorithm is designed and proposed, so as to further improve the effect and efficiency of problem solving. (3) To support the diffractive service composition approach, a SL-GABA algorithm is proposed for constraint decomposition.

This paper conducts performance comparison experiments on the proposed diffraction method based on priori correlation ABC, the experimental results show that the method proposed in this paper has better performance under different arbitration criteria than other ABC-based solving algorithms or other solving algorithms that take into account certain domain features. In addition, the influencing factors of the method proposed in this paper are studied through experiments.

Future works will involve proposing more flexible and fine-grained identification methods of core component services and diffraction-associated component services at each layer and constraint decomposition methods to adapt to more complex and challenging problem scenarios and user requirements. In addition, we will further analyze and explore the various elements and operation steps in ABC, so as to further improve the corresponding elements and operation steps. We will also apply the method proposed in this paper to more real-life service composition scenarios (such as intelligent old-age care, smart home, intelligent transportation and so on) to further verify the advantages of this method and its practical application values.

ACKNOWLEDGMENT

This work is supported by: The National Key Research and Development Program of China (Grant no. 2018YFB1402500), National Natural Science Foundation of China (Grant nos. 61872126 and 61832004).

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.

REFERENCES

- Ardagna, D., & Pernici, B. A. (2007). Adaptive service composition in flexible processes. *Proceedings of IEEE Trans. on Software Engineering*, 33(6), 369-384.
- Bin, P. (2005). *Knowledge and population swarms in cultural algorithms for dynamic environments* (PhD thesis). Wayne State University, Detroit, MI.
- Chifu, V. R., Pop, C. B., & Salomie, I. (2010). Selecting the optimal web service composition based on a multi-criteria bee-inspired method. *Proc. ACM Int. Conf. Information Integration and Web-based Applications and Services*, 40-47.
- Elfirdoussi, S., & Jarir, Z. (2019). An integrated approach towards service composition life cycle: A transportation process case study. *Journal of Industrial Information Integration*, 15, 138-146.
- Fekih, H., Mtibaa, S., & Bouamama, S. (2019). An Efficient User-Centric Web Service Composition Based on Harmony Particle Swarm Optimization. *International Journal of Web Services Research*, 16(1), 1-21.
- Haihong, E., Lin, Y., Song, M., Xu, X., & Zhang, C. (2019). A visual web service composition system based on process tree. In *2019 5th International Conference on Information Management (ICIM)* (pp. 274-278). IEEE. doi:10.1109/INFOMAN.2019.8714694
- He, J., Chen, L., Wang, X., & Li, Y. (2013). Web service composition optimization based on improved artificial bee colony algorithm. *J Netw*, 8, 2143-2149.
- Jang, J. H., Shin, D. H., & Lee, K. H. (2006). Fast quality driven selection of composite Web services. In *Proceedings of the 4th European Conf. on Web Services (ECOWS 2006)*. Institute of Electrical and Electronics Engineers Computer Society.
- Khanouche, M. E., Gadouche, H., Farah, Z., & Tari, A. (2020). Flexible QoS-aware services composition for service computing environments. *Computer Networks*, 166, 106982.
- Li, J., Fan, G., Zhu, M., & Yan, Y. (2019). Pre-joined semantic indexing graph for qos-aware service composition. *International Conference on Web Services (ICWS)*, 116-120.
- Liu, Z.-Z., Chu, D.-H., Jia, Z.-P., Shen, J.-Q., & Wang, L. (2016). Two-stage approach for reliable dynamic web service composition. *Knowledge-Based Systems*, 97, 123-143.
- Liu, Z. Z., Chu, D. H., Song, C., Xue, X., & Lu, B. Y. (2016, January). Social learning optimization(SLO) algorithm paradigm and its application in QoS-aware cloud service composition. *Inf. Sci.*, 326, 315-333.
- Ma, S.-P., Fanjiang, Y.-Y., & Kuo, J.-Y. (2014). Dynamic service composition using core service identification. *Journal of Information Science and Engineering*, 30(4), 957-972.
- Mezni, H., & Kbekbi, M. (2019). Reusing Process Fragments for Fast Service Composition: A Clustering-based Approach. *Enterprise Information Systems*, 13(1), 34-62.
- Niu, S., Zou, G., Gan, Y., Xiang, Y., & Zhang, B. (2019). Towards the optimality of QoS-aware web service composition with uncertainty. *International Journal of Web and Grid Services*, 15(1), 1.
- Osaba, E., Yang, X. S., Diaz, F., Lopez-Garcia, P., & Carballedo, R. (2016). An improved discrete bat algorithm for symmetric and asymmetric Traveling Salesman Problems. *Engineering Applications of Artificial Intelligence*, 48, 59-71.
- Peng, S., Wang, H., & Yu, Q. (2020). Multi-clusters adaptive brain storm optimization algorithm for qos-aware service composition. *IEEE Access: Practical Innovations, Open Solutions*, 8, 48822-48835.
- Salam, M.A., Bahgat, W.M., El-Daydamony, E., & Atwan, A. (2019). A novel framework for web service composition. *International Journal of Simulation-Systems, Science & Technology*, 20(3).
- Seghir, F., Khababa, A., & Semchedine, F. (2019). An interval-based multi-objective artificial bee colony algorithm for solving the web service composition under uncertain QoS. *The Journal of Supercomputing*, 75, 5622-5666.

Shirjini, M. F., Nikanjam, A., & Shoorehdeli, M. A. (2020). Stability analysis of the particle dynamics in bat algorithm: Standard and modified versions. *Eng. Comput.*, 1–12.

Thangaraj, P., & Balasubramanie, P. (2020). Meta heuristic QoS based service composition for service computing. *J. Ambient Intell. Hum. Comput.*, 1-7. 10.1007/s12652-020-02083-y

Wang, C., Ma, H., Chen, A., & Hartmann, S. (2019). Towards robust web service composition with stochastic service failures based on a genetic algorithm. In *Australasian Joint Conf. on Artificial Intelligence*. Springer.

Wang, H., Gu, M., Yu, Q., Tao, Y., Li, J., Fei, H., & Hong, T. (2019). Adaptive and large-scale service composition based on deep reinforcement learning. *Knowledge-Based Systems*, 180, 75–90.

Wang, X., Wang, Z., & Xu, X. (2013). An Improved Artificial Bee Colony Approach to QoS-Aware Service Selection. *Proc. Int'l Conf. Web Services*, 395-402.

Wang, Z., Cheng, B., Zhang, W., & Chen, J. (2020). Q-graphplan: QoS-aware automatic service composition with the extended planning graph. *IEEE Access: Practical Innovations, Open Solutions*, 8, 8314–8323.

Wu, Q., Zhu, Q., & Zhou, M. (2014). A correlation-driven optimal service selection approach for virtual enterprise establishment. *Journal of Intelligent Manufacturing*, 25(6), 1441–1453.

Xu, X., Liu, Z., Wang, Z., Sheng, Q. Z., Yu, J., & Wang, X. (2017). S-abc: A paradigm of service domain-oriented artificial bee colony algorithms for service selection and composition. *Future Generation Computer Systems*, 68, 304–319. doi:10. 1016/j.future.2016.09.008

Yaghoubi, M., & Maroosi, A. (2020). Simulation and modeling of an improved multi-verse optimization algorithm for QoS-aware web service composition with service level agreements in the cloud environments. *Simulation Modelling Practice and Theory*, 103, 1–16.

Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)* (pp. 65–74). Springer.

Yang, Y. F., Yang, B., & Wang, S. L. (2019). A dynamic ant-colony genetic algorithm for cloud service composition optimization. *International Journal of Advanced Manufacturing Technology*, 102(1/4), 355–368.

Zanbouri, K., & Jafari Navimipour, N. (2020). A cloud service composition method using a trust-based clustering algorithm and honeybee mating optimization algorithm. *International Journal of Communication Systems*, 33(5), e4259.

Zeng, L., Benatallah, B., & Ngu, A. H. H. (2004). QoS-Aware Middleware for Web Services Composition. *IEEE Transactions on Software Engineering*, 30(5), 311–327.

Zhou, J., Gao, L., Yao, X., Zhang, C., Chan, F. T. S., & Lin, Y. (2019). Evolutionary algorithms for many-objective cloud service composition: Performance assessments and comparisons. *Swarm and Evolutionary Computation*, 51(Dec), 100605.