Received March 1, 2020, accepted March 7, 2020, date of publication March 10, 2020, date of current version March 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979892

Multi-Clusters Adaptive Brain Storm Optimization Algorithm for QoS-Aware Service Composition

SHUNSHUN PENG[1,2], HONGBING WANG[1,2], (Member, IEEE), AND QI YU[3], (Member, IEEE)
[1]School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
[2]Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing 211189, China
[3]College of Computing and Information Sciences, Rochester Institute of Technology, Rochester, NY 14623, USA

Corresponding author: Hongbing Wang (hbw@seu.edu.cn)

This work was supported in part by the National Key Research and Development Plan under Grant 2018YFB1003800, in part by the NSFC Projects under Grant 61672152 and Grant 61532013, and in part by the Collaborative Innovation Centers of Novel Software Technology and Industrialization and Wireless Communications Technology. The work of Qi Yu was supported in part by the NSF IIS under Award IIS-1814450, and in part by the ONR under Award N00014-18-1-2875.

ABSTRACT Service composition provides an effective means to fulfill users' personalized requirements and complex business applications. With the number of web services rapidly growing, finding the best combination among services based on quality of service (QoS) poses a critical computational challenge due to the exponential growth of alternative composite services. Some efforts are developed to find the near-optimal service combination within an acceptable time range. They fall into two categories of solutions: exploring partial combinations in the service space and downsizing the optimization problem in scale. Although they solve the scalability problem to some extent, the required computational time is usually high. A promising direction is to integrate these two categories of solutions. However, its practical application suffers from three challenges: no good search scheme, no consideration of the natural organization of sub-problems, and the lack of diverse combinations. In this work, we propose a novel approach, called multi-clusters adaptive brain storm optimization (MCaBSO) algorithm. The proposed method uses brain storm optimization (BSO) as the search scheme to combine the division of search space with the exploration of the reduced search space. MCaBSO uses the twin support vector machine (TWSVM) to effectively divide the search space according to the natural organization of sub-problems. MCaBSO provides an adaptive dual strategy that gives guidance for the generation of diverse combinations. MCaBSO enables the agile exploration of the reduced search space and generates more high-quality combinations. MCaBSO is evaluated on two datasets to show effectiveness and efficiency.

INDEX TERMS QoS-aware service composition, brain storm optimization, twin support vector machine, adaptive dual strategy.

I. INTRODUCTION

Service composition has been used as an affordable and flexible means to integrate multiple existing services for on-demand software systems. These software systems gather services to form service-oriented systems, which provide agile development methodologies for many enterprises and flexible operations for users, respectively. QoS, such as price and reliability, can quantify the non-functional quality of services in many ways. It is typically used as an essential measure in service composition to select and compose web services. The goal of QoS-aware service composition is to find the best combination of web services.

The QoS-aware service composition optimization problem becomes especially important as the number of alternative composite services increases. As illustrated in Fig.1, there exists a business process with 5 tasks, and for each task there are 4 alternative web services with similar functionality but different QoS values. Then the number of alternative composite services will be 4^5. If the number of tasks or the number of alternative web services for each task increases, the number of alternative composite services grows exponentially. The exponential computation time for finding the best one from all the composite services is tractable only if the number of

The associate editor coordinating the review of this manuscript and approving it for publication was Nizar Zorba.

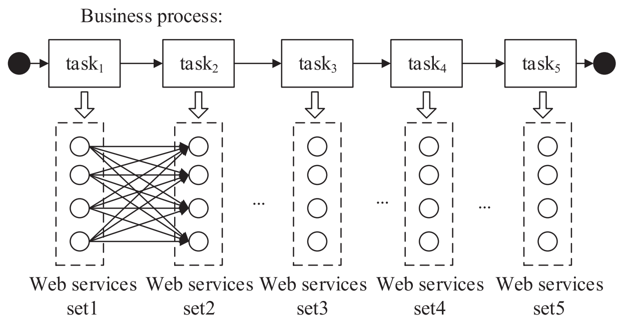


FIGURE 1. Process of the business example.

alternative composite services is small [1]. However, many enterprises and open infrastructures have adopted service-oriented architecture by decomposing their business applications into many smaller services [2]. For example, Netflix decomposes the online video rental application into services, which are hosted in a cloud provided by Amazon EC2. There are dozens of instance types for alternative services, such as m5.large [3]. The exponential response time will quickly become prohibitive to meet complex requirements.

To address this scalability problem, it is preferable to find a near-optimal combination at an acceptable cost. Existing solutions fall into two categories. The first category relies on the exploration of a subset of combinations in the service space [1], [4]–[8]. They prune the candidate services based on the QoS comparison between one service and other functionally-equivalent services, or guide the exploration towards the optimum. The second category transforms the optimization problem into several sub-problems [9]–[11]. Although these approaches solve the scalability problem to some extent, the required computational time is usually high.

In this paper, we aim to combine the key ideas from the existing two categories of approaches to find an optimal or near-optimal combination in the shortest time. In particular, we will tackle the following challenges. First, there is no good search scheme to combine these categories of solutions. A poor scheme can not take advantage of different categories of solutions, even affect their effectiveness. Second, the division of the optimization problem does not consider the natural organization of sub-problems, which might lead to the overlapping and missing of the search space. The overlapping of the search space would increase the response time for a request, while the missing of the search space would reduce the optimization effect. Third, maintaining diverse composite services is still needed to provide high-quality combinations. The various combinations are important for service composition, which can effectively avoid local optimization.

For the first problem, we use the evolution process of BSO as the search scheme. BSO [12] is a new intelligent optimization method based on principles of the brainstorming process. It uses the clustering strategy to transform the optimization problem into multiple local ones, and applies evolutionary operators with a disturbance to explore combinations. The intertwined process of convergence and divergence makes BSO suitable for the cooperation of the problem reduction and the exploration of the search space. For the second problem, we use TWSVM to partition the optimization problem into several local ones. TWSVM [13], [14] constructs nonparallel planes, around which the combinations of the corresponding class are clustered. TWSVM simplifies the classification problem and well organizes the individual clusters. For the third problem, an adaptive dual strategy is employed to generate more high-quality combinations. The intra-cluster operator generates new combinations by exploiting the QoS value of combinations and the closeness of the combinations with the cluster, while the inter-cluster operator makes new combinations by exploring the search space. The adaptive dual strategy adopts dynamic parameters to adjust the usages of these two operators in different stages. It can explore farther areas in the search space and exploit more optimal information.

In this paper, we name our method, multi-clusters adaptive brain storm optimization (MCaBSO) algorithm. It is a novel hybrid approach based on BSO, TWSVM, and adaptive dual strategy. The contributions of this paper are summarized below.

1) We propose a hybrid approach, called MCaBSO, adopting BSO as the basic search scheme, TWSVM as the problem reduction strategy, and an adaptive dual strategy as the guidance for the generation of high-quality combinations.
2) On-the-fly search space exploration- MCaBSO divides the search space into several small search spaces, according to the natural organization of sub-problems. Furthermore, MCaBSO learns the QoS information of different combinations in one small search space and the closeness between combinations to the corresponding class. The data can be used to help the combinations converge into a small area.
3) Novel strategy for high-quality combinations- MCaBSO employs an adaptive dual strategy to leverage the advantages of the intra-cluster and inter-cluster operators in different stages, so as to explore more high-quality combinations.

We have conducted the evaluation of MCaBSO on real-world data, which shows that MCaBSO spends a shorter time on finding the optimal solution in comparison with existing approaches. Besides, MCaBSO exhibits good convergence speed even with the increasing numbers of tasks and concrete services. Therefore, MCaBSO can effectively and efficiently solve the scalability issue. The remainder of the paper is organized as follows. Section II gives an overview of the related work. Section III introduces the QoS-aware service composition model. Section IV presents the basic BSO and TWSVM. Section V describes the proposed MCaBSO in detail. Section VI shows some experimental results to evaluate our approach. Section VII summarizes the paper and outlines future directions.

II. RELATED WORK

In this section, we provide an overview of the related work on BSO and QoS-aware service composition that are most relevant to the proposed approach.

A. BRAIN STORM OPTIMIZATION

Since BSO was proposed in 2011, some improvements have been made on both the algorithm side and the application side. A novel BSO extension was developed in [15] to increase population creativity. Two re-initialization individuals strategies are applied to create more possibility for various individuals, which improve the global optima. Yang *et al.* [16] propose an advanced discussion mechanism to enhance the diversity of BSO. Meanwhile, a differential strategy is applied to accelerate the convergence rate. Peng *et al.* [17] integrate BSO with pbest guided step-size for complex optimization problems. A self-adaptive strategy is applied to solve the dependence of original BSO on several control parameters. Meanwhile, a dynamic clustering number strategy is used to converge the algorithm to an optimal solution quickly. The authors in [18] improve the performance by integrating the fuzzy min-max neural network with BSO. The feature extraction on a subset of data improves predictive accuracy and reduces the search space.

BSO can be applied in a broad range of scenarios. The work of [19] proposes a novel BSO in the electromagnetic field to optimize the brushless DC wheel motor. The evolutionary operators (e.g., clustering) extract global information and then improve diversity. In the same area, the work of [20] also proposes an enhanced BSO method for electromagnetic application. Ma *et al.* [21] solve the wind speed forecasting by using BSO to improve the fuzzy neural network. The work in [22] develops BSO to solve the optimization problem of UAV formation flight. Sato and Fukuyama [23] introduce a global-best BSO into a smart city. The method includes the total optimization of energy costs, CO_2 emission minimization and electric power. The work in [24] applies the brain storm optimization graph theory into the medical image field for specifying energy computation mapping. Hao *et al.* [25] develop a hybrid brain storm optimization approach to solve distributed hybrid flowshop scheduling problems.

B. QoS-AWARE SERVICE COMPOSITION

A wide range of approaches have been developed to improve the performance of service composition as the number of services grows, including the selection of representative services, the exploration of the partial combination candidates and the transformation of optimization problem into several sub-optimization problems.

Some approaches based on selecting representative component services achieve search space reduction, since the services that are verified impossible to participate in the optimal combination are pruned. The work of [1] provides candidate services prune based on QoS comparison, and then select services with better QoS for the service composition. Li *et al.* [26] select representative services according to the QoS correlation among services and QoS correlations of user requirements. The work of [27] uses a cross-layer based dynamically tuned queue length scheduler, to guarantee QoS. In [28], the authors propose a time-series-based prediction method to select representative services for service composition. The work in [29] proposes an optimization approach to select a set of promising services dynamically. A Directed Acyclic Graph describes the semantics of service composition. Then each execution path is extracted and the optimal global solution is obtained by solving the mixed integer linear programming problem. Tan *et al.* [5] reduce the search space based on local optimality of the services and global constraints. However, these approaches would cost too long when the candidate services set becomes larger.

These approaches, based on exploring the partial combination candidates, also reduce the search space. The work of [7] proposes a hybrid approach to solve the optimization problem. The proposed method reduces alternative combinations by chained dynamic programming. Khanouche *et al.* [30] make use of the k-means method to handle the candidate services into different QoS levels. According to the resulting clusters, the composite service is generated by composing the promising candidate services. The unpromising candidates are filtered out to improve the efficiency of service composition. To reduce the cost, Gavvala *et al.* [31] propose a metaheuristics algorithm to approximate an optimal solution. The authors make use of the Whale Optimization Algorithm to generate the composition candidates, and manipulate them using genetic operators for evolving new candidates. Additionally, the eagle strategy is used for taking a balance between the search space exploration and the population information exploitation. The work of [32] proposes a novel ant colony optimization algorithm to find the best combination. In [33], the authors propose an abstract-refinement method to guide the exploration of the complete composition graph. Wang *et al.* [34] combine graphplan with the heuristic search algorithm to improve search effectiveness. These approaches reduce the number of candidates effectively, but the problem scale is still massive, so that the required computational time is high.

To further improve efficiency, approaches based on transforming the optimization problem into several sub-optimization issues have been proposed. Zhou and Yao [10] develop a multi-population differential artificial bee colony algorithm to downsize the problem scale. Hossain *et al.* [11] submit a two-phase approach to find an optimal or near-optimal solution. The proposed method combines the parallel processing with the selection of candidate services. The paper [35] proposes a parallel method based on MapReduce to find the top-k compositions. The central agent divides service composition into several mutually independent tasks, and then multiple agents execute them in parallel for reducing time. Besides, this method selects services with better QoS to avoid exhaustive compositions while finding a set of optimal solutions. The work of [36] proposes a parallel refined probabilistic approach to improve performance. Multiple agents work in parallel to speed up the convergence and an elegant probabilistic model is constructed to guide the optima. However, the division of the optimization problem does not consider the natural organization of subproblems.

In this work, our focus is to optimize the QoS-aware service composition effectively and efficiently. MCaBSO is proposed to utilize BSO for combining the problem reduction with the exploration of combinations in services space. It also divides the problem according to the natural organization of subproblems and provides an adaptive dual strategy to adapt to the state of the search space. Therefore, MCaBSO can reduce the optimization problem into several refined subsets optimization problems and improve the optimality.

III. QoS-AWARE SERVICE COMPOSITION MODEL

In this section, we focus on the quality of service that can be quantified using QoS metrics. Then the optimality function for service composition is given. The problem of QoS-aware service composition is declared in the end.

A. QUALITY OF SERVICE

We herein introduce the QoS of component service and composite service, respectively.

The QoS of component service is reflected through the QoS attributes of service, including two main classes: one is the positive attributes (e.g., throughput), and the other is negative one (e.g., response time). The positive QoS attributes have a positive impact on the non-function of service, while the negative attributes have a negative effect on the non-function of service. Therefore, the positive QoS attribute and negative QoS attribute values need to be maximized and minimized, respectively.

Since the QoS of a composite service is aggregated from the QoS of its component services, its QoS aggregation is impacted not only by the QoS attributes of component services, but also by the compositional structures that determine how to connect different component services. The most basic compositional structures are sequential, parallel, loop, and conditional. The sequential composition is the simplest, as long as the corresponding services $\{s_1, s_2, \ldots s_n\}$ are executed one by one according to the order of solving problems. The parallel composition aims to make more services $\{s_1, s_2, \ldots s_n\}$ run at the same time. The loop composition needs to execute service s_i k times repeatedly. There are different directions according to the evaluation of the guard conditions, conditional composition $\{s_1, s_2, \ldots, s_n\}$ is the structure that chooses one direction to deal with service s_i. They determine the QoS of composite service by summing the QoS values of component services, multiplying the QoS values of component services and so on.

Under these four basic structures, considering the aggregation functions of the response time and throughput can almost cover elementary types of QoS aggregation function, we select them as the examples. Their QoS aggregation functions are shown in Table 1. $q(s_i)$ represents the standardized value of QoS attribute for s_i. The response time is the time from sending a service call request to receiving a response. During the sequential execution, the value is computed by summing the response time of n services. During the parallel execution, the maximum response time among n services is the response time of the composite service. During the loop execution, the value is obtained by summing the response time of the involved service k times. During the conditional execution, the maximum response time among n services is the response time of the composite service. The throughput is the total number of invocations processed by the service per second.[1] During the sequential execution, the minimum throughput in all component services determines the throughput of the composite service. During the parallel execution, it needs to sum the throughput of n component services. During the loop execution, it is computed by determining the throughput of the involved service. During the conditional execution, the minimum throughput among n component services is selected as the throughput of the composite service.

TABLE 1. Examples of QoS aggregation functions.

Attributes	Sequential	Parallel	Loop	Conditional
response time	$\sum_{i=1}^{n} q(s_i)$	$\max_{i=1}^{n} q(s_i)$	$k * q(s_i)$	$\max_{i=1}^{n} q(s_i)$
throughput	$\min_{i=1}^{n} q(s_i)$	$\sum_{i=1}^{n} q(s_i)$	$q(s_i)$	$\min_{i=1}^{n} q(s_i)$

B. OPTIMALITY ASSESSMENT

Consider that the optimality is evaluated from multiple QoS attributes angles, Simple Additive Weighting (SAW) technique [37] is applied to compute a score for the attributes vector. Since the values of the attributes have different units and ranges, the first step of SAW is to normalize the values of QoS attributes into the range (0, 1). The normalization is implement by comparing with the maximum and minimum QoS attribute value in a service class, which is given as follows. For the positive attributes, we have the normalization:

$$q^i(s) = \begin{cases} \dfrac{at^i(s) - at^i_{min}(s)}{at^i_{max}(s) - at^i_{min}(s)}, & at^i_{max}(s) \neq at^i_{min}(s) \\ 1, & at^i_{max}(s) = at^i_{min}(s) \end{cases} \tag{1}$$

For the negative attributes, we have the normalization:

$$q^i(s) = \begin{cases} \dfrac{at^i_{max}(s) - at^i(s)}{at^i_{max}(s) - at^i_{min}(s)}, & at^i_{max}(s) \neq at^i_{min}(s) \\ 1, & at^i_{max}(s) = at^i_{min}(s) \end{cases} \tag{2}$$

where $at^i(s)$ is the i-th QoS attribute value for service. $at^i_{max}(s)$ and $at^i_{min}(s)$ are maximum and minimum values for i-th attribute in service class. According to the composite structure, the aggregation function of i-th attribute for the composite service is computed. Table 1 lists a part of formulas.

[1]https://qwsdata.github.io/

Due to the user's preference, the next step of SAW is to embody the importance of different QoS attributes. The fitness value of the composite service cs is computed as follows.

$$FV(cs) = \sum_{i=1}^{n} f^{i}(cs) w^{i}(cs) \tag{3}$$

where $w^{i}(cs)$ is the weight of i-th attribute for cs, and $f^{i}(cs)$ is the aggregation function of i-th attribute for cs.

According to the score for the attributes vector, we can assess the optimality by comparing the size of the score. The optimal composite service must be selected among the feasible composite service. The feasible composite service and optimal composite service are defined as follows.

Definition 1 (Feasible Composite Service): Let $\boldsymbol{GC} = \{c_1, c_2, \ldots c_n\}$ be a vector of QoS constraints on user requirements. Let composite service cs be composed by the concrete services, which are selected from different service classes. cs is a feasible composite service if i-th QoS aggregation value of cs does not exceed the c_i.

Definition 2 (Optimal Composite Service): An optimal composite service is a composite service that its optimal global value is better than that of other feasible composite services.

C. PROBLEM DESCRIPTION

There are more than one mapping from abstract composite service to concrete combinations, the problem of finding the best combination can be seen as an optimization problem. To solve it, the most direct way is to obtain all possible combinations, and then select the combination with maximal QoS value. However, as the number of abstract services and optional concrete services increases, the number of combinations increases exponentially. This way of finding the best combination would take a lot of time and resources. As web service composition needs to be handled at runtime, users are more willing to accept a near-optimal combination with lower cost, rather than an exact one with higher cost [5].

A favorite method is to downsize the problem scale and explore the partial search space under the guidance of optimal information. This approach is particularly helpful for handling a large number of combinations. However, the questions that arise are not only how to divide the search space, but also how to guide the exploration of the reduced search space. In this work, we will focus on optimizing QoS-aware service composition efficiently and effectively by converging combinations to a refined area that has a small exploration area and is likely to find an optimal or near-optimal combination.

IV. PRELIMINARIES OF THE APPROACH

BSO is new intelligent optimization method based on principles of the brainstorming process [12]. A group of people with different backgrounds will get together to generate great ideas, which can be seen as solution candidates to a specific problem. The clustering strategy and evolution operators are applied to these solution candidates in such a way to simulate better solutions. The selection operation is used to pass the elites to the next generation. The application of BSO to solve the optimization problem has been proposed by several researchers [20], [38]. As illustrated in Fig.2, a BSO starts with the randomly generated ideas to a specific problem, which we call them initial individuals. Clustering operators are applied on these individuals to create different clusters. Creation operators are utilized to create new individuals by mutating selected individuals or interacting with more than two individuals. After new individuals are generated, the better one will be selected as an individual in the new generation when compared with the one waiting for an update. The selection is operated based on the fitness value of an individual, where the highly-fit individual has a higher chance to be selected into the next iteration. The fitness value of an individual quantifies the performance quality from multiple QoS attributes perspective. When the termination condition is satisfied, the brainstorming process will terminate. For example, BSO would stop when the number of iterations is higher than the value set in advance.

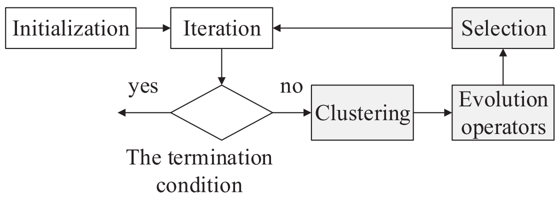


FIGURE 2. The workflow of BSO.

TWSVM [13], [39] is an efficient classification method based on the statistical learning theory. It has some theoretical and computational traits, such as novel small sample learning, high generalization ability, unique global solution, and two nonparallel proximal hyperplanes. These traits simplify the classification problem, reduce the computational complexity, and provide more useful guidance on solution discovery. Therefore, TWSVM has been applied to data classification problems effectively. For the multi-classification problem, we induce it by constructing multiple hyperplanes. It implements the following idea: the input vectors are mapped to high-dimensional feature spaces. In the feature spaces, multiple hyperplanes are sought so that each hyperplane is as close as possible to the points of one cluster and keeps away from the points of other clusters as far as possible.

V. THE MCaBSO FOR SERVICE COMPOSITION

Our work proposes a hybrid approach for QoS-aware service composition. This section outlines the proposed MCaBSO and details each algorithmic component.

A. THE OVERVIEW OF MCaBSO

We propose our approach, MCaBSO, to support the on-the-fly exploration of the search space and guide the individuals' selection towards the optimum. Fig.3 gives the workflow of MCaBSO. First, MCaBSO randomly constructs a set of

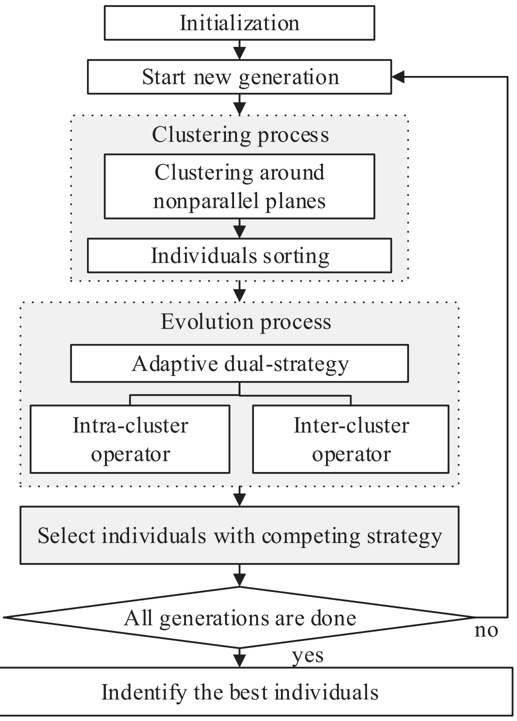


FIGURE 3. The workflow of the proposed algorithm.

individuals, where the web services are picked randomly from each set of web services that corresponds to each abstract service class. MCaBSO then partitions the individuals into several clusters so that the optimization problem is transformed into multiple local optimization problems. The individuals in one cluster are sorted according to their QoS values and the closeness between them within the cluster. Then, an adaptive dual strategy is used to adjust the usages of intra-cluster and inter-cluster operators for generating new individuals. Eventually, the selection operator with a competing strategy provides sufficient confidence to pin down the desired combinations. The genetic operators with different strategies will be introduced in the rest of this section.

B. CLUSTERING PROCESS

The first phase is to partition the individuals into clusters, so that the optimization problem is divided into multiple local optimization problems. Then, the individuals of one cluster are sorted.

1) CLUSTERING AROUND NONPARALLEL PLANES

The TWSVM method is to seek multiple nonparallel planes so that each plane is as close as possible to the individuals of one cluster and keeps away from the individuals of other clusters as far as possible. The geometric presentation of TWSVM is shown in Fig.4. The hollow circles, rectangles and solid circles representing the individual points, are respectively divided into three clusters by planes $plane_1(x)$, $plane_2(x)$ and $plane_3(x)$. They are as close as possible to their planes. Meanwhile, the hollow circles stay far away

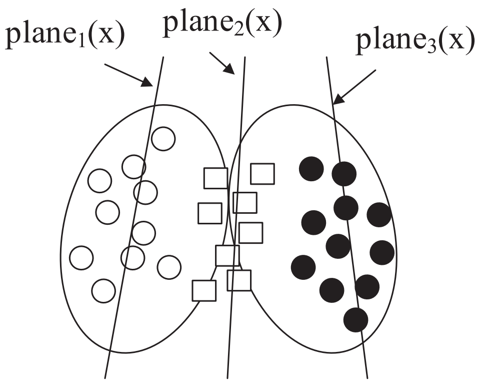


FIGURE 4. Geometric presentation of TWSVM.

from $plane_2(x)$ and $plane_3(x)$, the rectangles stay away from $plane_1(x)$ and $plane_3(x)$, the solid circles stay away from $plane_1(x)$ and $plane_2(x)$.

We consider m individuals $\{cs_1, cs_2, \ldots, cs_m\}$. The individuals around one plane is clustered according to the distance from the individual point to the corresponding plane. The distance is measured according to their QoS values. Assuming n QoS attributes of an individual cs_i are concerned, there is a n-dimension vector $\boldsymbol{Q}(cs_i) = < f^1(cs_i), f^2(cs_i), \ldots, f^n(cs_i) >$ to represent the QoS vector of cs_i. The QoS values of m individuals in the n-dimensional vector space $\boldsymbol{R}^n$ can be represented by a $m \times n$ matrix $\boldsymbol{Q}$.

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}(cs_1) \\ \boldsymbol{Q}(cs_2) \\ \cdots \\ \boldsymbol{Q}(cs_m) \end{bmatrix} = \begin{bmatrix} f^1(cs_1) & f^2(cs_1) & \ldots & f^n(cs_1) \\ f^1(cs_2) & f^2(cs_2) & \ldots & f^n(cs_2) \\ \cdots & & & \\ f^1(cs_m) & f^2(cs_m) & \ldots & f^n(cs_m) \end{bmatrix} \tag{4}$$

where $f^j(cs_i)$ represents the aggregated value of j-th QoS attribute of cs_i. The QoS aggregation functions of the response time and throughput are listed in Table 1.

These individuals are partitioned into K clusters by K nonparallel planes, which are represented as follows.

$$\begin{aligned} plane_1(x) &= \omega_1 x + b_1 \\ plane_2(x) &= \omega_2 x + b_2 \\ &\cdots \\ plane_K(x) &= \omega_K x + b_K \end{aligned} \tag{5}$$

where $k \in \{1, 2, \ldots, K\}$ labels the cluster to which the i-th individual belongs; $\omega_k \in \{\omega_1, \ldots, \omega_K\}$ and $b_k \in \{b_1, \ldots, b_K\}$ are the parameters of $plane_k(x)$.

To determine these nonparallel planes, the parameters ω_k and b_k need to be determined from the following formula:

$$\begin{aligned} &\min_{\omega_k, b_k, \xi_k} \frac{1}{2} \|\boldsymbol{X}_k \omega_k + b_k \boldsymbol{e}\|^2 + c\boldsymbol{e}^\tau \boldsymbol{\xi}_k \\ &subject\ to\ \hat{\boldsymbol{X}}_k \omega_k + b_k \boldsymbol{e} \geq \boldsymbol{e} - \boldsymbol{\xi}_k \end{aligned} \tag{6}$$

where $\boldsymbol{X}_k$ represents the QoS vectors of the individuals belonging to the k-labeled cluster, $\boldsymbol{e}$ denotes a vector of ones with appropriate dimension; c is the penalty parameter that takes a balance between the distance and the correctness of division; $\boldsymbol{\xi}_k \in \boldsymbol{R}^{m-m_k}$ is a slack vector; $\hat{\boldsymbol{X}}_k$ refers to QoS vectors of the individuals belonging to other clusters.

To optimize Equation (6), the Lagrange function is introduced to obtain its Wolfe dual as follows.

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{G}(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{G}^T\boldsymbol{\alpha} - \boldsymbol{e}^T\boldsymbol{\alpha} \tag{7}$$

where $\boldsymbol{H} = (\boldsymbol{X}_k, \boldsymbol{e})$, $\boldsymbol{G} = (\hat{\boldsymbol{X}}_k, \boldsymbol{e})$, and $\boldsymbol{\alpha}$ represents the vector of Lagrangian multipliers.

$[w_k, b_k]^T$ can be obtained from the solution of Equation (7) by the Karush-Kuhn-Tucker condition:

$$[w_k, b_k]^T = (\boldsymbol{H}^T, \boldsymbol{H})^{-1}\boldsymbol{G}^T\boldsymbol{\alpha} \tag{8}$$

Then, the plane of the k-labeled cluster can be obtained by:

$$plane_k(x) = w_k x + b_k \tag{9}$$

Based on K nonparallel planes, cs is labeled according to the minimum distance in terms of its QoS values. The marking problem can be formally specified as follows.

$$plane(cs) = \min_{k=1,2,\ldots,K} \{d_k\} \tag{10}$$

where d_k represents the distance from cs to the k-labeled hyper plane. It is computed as follows.

$$d_k = \frac{|\omega_k \boldsymbol{Q}(cs) + b_k|}{||\omega_k||} \tag{11}$$

where $|\cdot|$ represents the absolute value and $||\omega_k|| = \sqrt{\sum_{i=1}^{k} \omega_i^2}$.

2) INDIVIDUALS SORTING

After dividing the individuals into clusters, the next step is to sort the individuals for each cluster according to individuals' quality. As we discussed in Section III-B, the quality is reflected through the fitness value that is calculated based on multiple QoS attributes and users' preferences. Nevertheless, it does not provide a direct indication on how likely the individual can belong to a cluster. To address this problem, we propose an estimation on the closeness of the individual with a cluster with a certain QoS level. Firstly, we introduce the notion of cluster midpoint whose fitness value represents the QoS level of the cluster. Assume that the k-labeled cluster has r individuals $\{cs_1^k, cs_2^k, \ldots, cs_r^k\}$, we calculate the fitness value of the cluster midpoint CM^k as follows.

$$FV(CM^k) = \max_{i=1}^{r} FV(cs_i^k) \tag{12}$$

For K clusters, we recursively apply Equation (12) to calculate the fitness value of each cluster midpoint. Subsequently, the closeness of the individual cs_i^k with the k-labeled cluster is given.

$$Clo(cs_i^k) = \frac{FV(cs_i^k)}{FV(CM^k)} \tag{13}$$

In Equation (13), the closeness of the individual that belongs to the cluster with the QoS level $FV(CM^k)$ would increase with the growth of the individual' QoS. In other words, the individual with lower QoS value in the cluster may has a lower chance of being selected for guiding the best exploration.

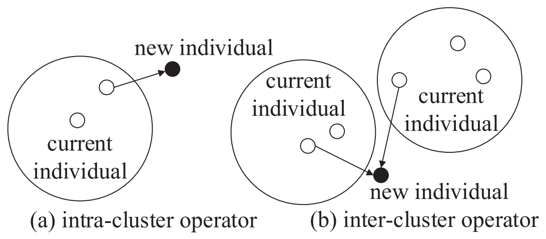


FIGURE 5. Two operators to create new composite services.

Given the fitness value and closeness, the new utility is proposed to increase the service composition optimization and reduce the search space of solutions. The utility is formally defined.

$$Utility(cs) = Clo(cs)FV(cs) \tag{14}$$

where the term Utility(cs) represents the worth of the individual.

After obtaining the sorted individuals' list, high-quality individuals are preferred as parents to generate new ones. It not only refines the information of the solutions, but also reduces the deletion of promising solutions. For example, the cluster with different QoS levels features the quality of the solutions, and the term fitness evaluation distinguishes the solutions in the same cluster. When the fitness values of solutions are low, the relevance term increases the probability of these solutions contributing to the evolution of the next generation.

C. EVOLUTION STAGE

In the evolution stage, the intra-cluster and inter-cluster operators take the abstract services of high-quality individuals as the operational dimension to generate new individuals. An adaptive dual strategy is also brought to get the most out of these two operators at different evolutionary stages.

1) INTRA-CLUSTER AND INTER-CLUSTER OPERATORS

The intra-cluster and inter-cluster operators are displayed in Fig.5. In Fig.5(a), the intra-cluster operator creates new individuals by changing current individuals randomly. While in Fig.5(b), the inter-cluster operator creates new individuals by mutating the mixture of two current individuals coming from different clusters. For the inter-cluster, usually two clusters are used to create new individuals since three or more clusters would increase the complexity of the algorithm [12].

These two operators follow similar steps as those from the original BSO, but different in the operational dimension. Instead of taking the decision variable as a dimension, we use each abstract service as a dimension and assign it the value of identifying a candidate service of the abstract service. In this dimension space, an individual is represented as $cs =< id(as_1), id(as_2), \ldots, id(as_D) >$, where $id(as_d)$ is the index of the candidate service for the d-th abstract service as_d, and D is the number of abstract services.

For these two operators, there is a disturbance for improving creativity, because their performance critically relies on the local optima value of current individuals. Consider a set of individuals CS $= [cs_1, cs_2, \ldots cs_m]$ composing of m individuals. In the first operator, new individual $cs'_i =< id'_i(as_1), id'_i(as_2), \ldots, id'_i(as_D) >$ is generated by adding a disturbance on a component service of cs_i as follows.

$$id'_i(as_d) = id_i(as_d) + round(rand(-1, 1) \times (id_i(as_d) - id_m(as_d))) \quad (15)$$

where rand() is the random values function and round() is the rounding function. In this operator, the $id'_i(as_d)$ may be out of the candidate services set. In such a situation, the value would be adjusted as the corresponding boundary value.

In the second operator, the disturbance is added to the combination of two individuals from different clusters. The first step is to cross two individuals cs_i and cs_j, which come from different clusters k_i and k_j. The component services of cs_i are substituted by the component services of cs_j as follows.

$$id'_i(as_d) = id_j(as_d) \quad (16)$$

Based on the crossed individuals, the second step is to add the disturbance. The new cs' is generated by Equation (15).

In the process of creating new individuals, the intra-cluster operator is around one current individual in the cluster. It is easy to use the local information of the cluster to refine the search region, which enhances the exploitation ability. On the contrary, the inter-cluster operator is implemented on two clusters, which may get possible information from more clusters and keep away from the existing individuals. It is easy to explore a larger area, which enhances the exploration ability.

2) ADAPTIVE DUAL STRATEGY

In the initial stage of evolutionary iteration, there is a lot of search space to explore for finding possible optimal individuals. With the evolution of MCaBSO, essence information is retained to the next generation and the solutions space is continuously being refined. More attention is focused on the exploitation of essential information to find a global and local optimum. Considered the advantages of the intra-cluster and inter-cluster operators, we need to make a trade-off between them. This is to say, we need to balance the utility of inter-cluster and intra-cluster operators. The adaptive dual strategy is applied to control the contribution degree of the two operators in different stages of evolution.

In this strategy, the probability value is used to measure the chance of using one operator. The probability p_{inter} of using inter-cluster should be high enough for exploring more possible global individuals in the early stage, while low enough for exploring as much as possible search filed in the later stage. The probability p_{intra} of using intra-cluster should be low enough for exploiting the information in the early stage,

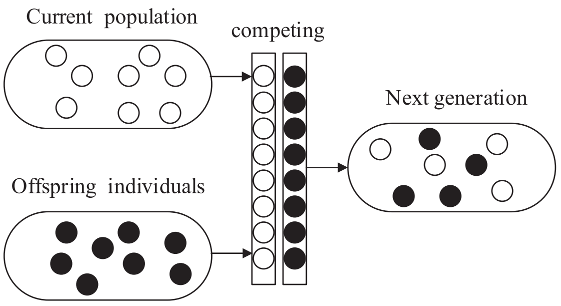


FIGURE 6. Selection process of MCaBSO.

while high enough for refining the local optimum in the later stage. They are calculated as follows.

$$p_{inter} = p_{low} + p_{high} \times (1 - \frac{N_{Cgen}}{N_{Mgen}}) \quad (17)$$

$$p_{intra} = 1 - p_{inter} \quad (18)$$

where p_{low} and p_{high} are two constants that are used to determine the proportions of using inter-cluster operator and intra-cluster operator, respectively. N_{Cgen} and N_{Mgen} are the current and maximum iteration numbers.

As mentioned above, N solutions can be created by the creation operators with the adaptive dual strategy.

D. OPTIMAL SELECTION STAGE

After new individuals are generated by intra-cluster and inter-cluster operators, the selection operator is performed to retain these individuals that are likely to contribute to high optimal value to the next generation. To overcome shortcomings resulting from the centralization of population selection, such as immature convergence and fast convergence speed, we introduce an Enhanced Selection Policy (ESP) to increase the probability of rapidly converging solutions to an optimal value.

The ESP is introduced in Fig.6. Suppose there now have eight individuals in the current generation $cg =< cs_1, cs_2, \ldots, cs_8 >$ and also eight offspring individuals $ng =< cs'_1, cs'_2, \ldots, cs'_8 >$ generated through creation operators, we select a pair of individuals (e.g., cs_3 and cs'_4) randomly from these two populations, respectively. A competing strategy is applied to the pair of individuals to realize the selection of the favored combinations in the next generation. The competing strategy is a plan for directing and managing the actions of cs_3 and cs_4. Considering the QoS attributes, it can be specified as a comparison with the global optimality values of the two composite services. The global optimality is measured by the utility of composite service, which is calculated using Equation (14) with QoS attributes normalized using Equation (1) or Equation (2). After each competition, some better solutions having better utility values will be selected for the next generation.

The selection policy would reduce the evaluation computation of comparing the utility value of each individual with that of all other individuals. Also, the right individuals that choose the better utility value from a pair of individuals would be retained.

VI. EXPERIMENTAL AND ANALYSIS

In this section, we present our experimental evaluation to demonstrate the performance of the proposed MCaBSO algorithm. All experiments are conducted on a 3.4GHz PC with 8GB RAM.

A. EXPERIMENT DESIGN

1) DATASETS

We use the following two publicly available datasets in our experiments:

- QWS dataset[2] [40]: This dataset is collected from public sources on the web, consisting of 2,507 web services and 9 QoS attributes.
- WS-DREAM dataset2[3] [41]: This dataset is obtained by crawling web service information on the web. It contains 1,974,675 real-world web service invocation information on 5,825 real-world web services. Each record contains the response time and throughput attributes of a web service.

The composite service is evaluated by the fitness value, which compromises the values of aggregation functions of different QoS attributes into a score. Different QoS attributes might have similar aggregation functions, for example, the cost attribute has a similar maximum function with the response time in the conditional structure. Therefore, we only select representative aggregation functions. Considering that the aggregation function of response time and throughput can cover the maximum function, minimum function, summation function and so on, we choose to use response time and throughput attributes. One of the main objectives of the experiments is to evaluate the efficiency of the proposed approach in solving the scalability issue. The QoS constraints are set to make the service composition scenario more believable. The constraints with more restriction would filter too many services so that the experiment cannot effectively evaluate the proposed approach in an inappropriate scenario. In consideration of a believable and appropriate service composition scenario, the threshold values should be smaller. According to the method of fixed percentage [5], the threshold values are set as 10% of the difference between the maximum value and the minimum value bigger than the minimum value. Moreover, since the values of response time and the throughput are normalized, we use similar formulas to set the threshold values as follows.

$$c_{rs} = q_{min}^{rs} + \frac{q_{max}^{rs} - q_{min}^{rs}}{10}$$

$$c_{th} = q_{min}^{th} + \frac{q_{max}^{th} - q_{min}^{th}}{10}$$

where q_{min}^{rs} and q_{max}^{rs} denote the minimum and maximum normalized response time, respectively; q_{min}^{th} and q_{max}^{th} denote the minimum and maximum normalized throughput, respectively.

[2]https://qwsdata.github.io/

[3]http://inpluslab.com/wsdream/

2) COMPARISON METHODS

To demonstrate the effectiveness of the proposed approach, we compare with the following four related approaches:

- BSO: The brain storm optimization with the k-means proposed in [12].
- BSO-kPC: The brain storm optimization with the k-plane proposed in [42].
- GA-HS: The improved genetic algorithm proposed in [43].
- PBA: Partition-based artificial bee colony algorithm [8].

The reason for adding BSO and BSO-kPC is because MCaBSO is also taking peoples' brainstorming as the problem-solving prototype. Using the same prototype will allow BSO, BSO-kPC and MCaBSO to gather group wisdom to find the optima, therefore enable us to compare the effect of the clustering strategies. As for choosing GA-HS and PBA, they are also optimization algorithms based on swarm intelligence. Similar to GA-HS and PBA, MCaBSO also takes into consideration the time cost and optimization. A similar motivation enables us to compare the effectiveness and efficiency of the three algorithms.

3) EVALUATION METRICS

We evaluate the five algorithms in terms of both effectiveness and efficiency.

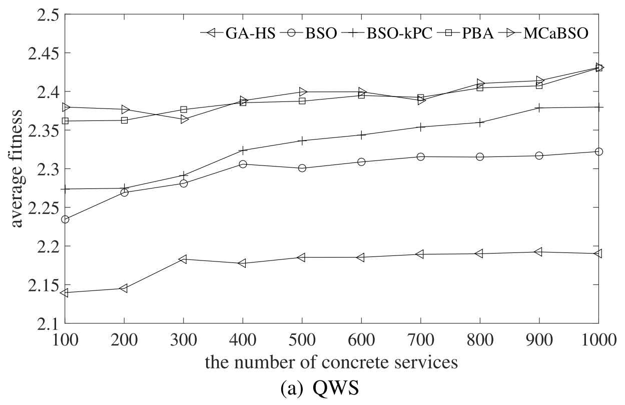
a: EFFECTIVENESS

To evaluate the effectiveness, we use the concepts of the average fitness and best fitness. The fitness is the aggregate QoS value of a composite service with the specific instances of multiple abstract services given by Equation (3). The average fitness is to average all solutions' fitness values after 200 generations in one experiment. It helps evaluate the effectiveness in terms of the diversity of solutions. The best fitness is to maximize all solutions' fitness values after 200 generations. It helps evaluate the effectiveness from the angle of the optimal scheme.

b: EFFICIENCY

To evaluate the efficiency, we compute the convergence time of the five algorithms.

The values of evaluation metrics would vary with various configurations. Two specific configurations are shown in Table 2. In scenario#1, each service request has 5 abstract services. The number of concrete services for each abstract service increases from 100 to 1000. In scenario#2, there are 100 concrete services for each abstract service. The number of abstract services increases from 5 to 50.



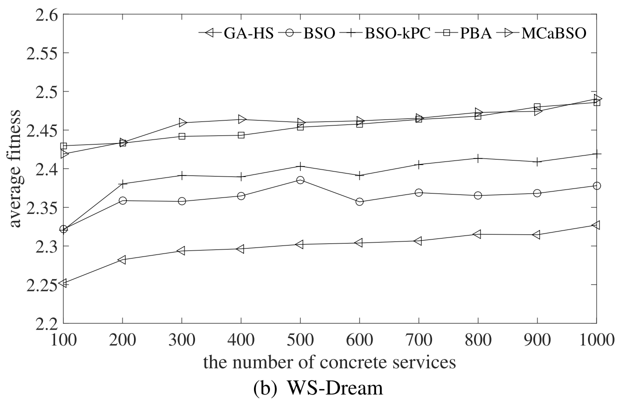


FIGURE 7. Average fitness in scenario#1.

TABLE 2. Scenarios configuration.

Factor	Scenario#1	Scenario#2
Number of abstract services	5	from 5 to 50
Number of concrete services	from 100 to 1000	100

TABLE 3. Average fitness for QWS in scenario#2.

Abstract Services	GA-HS	BSO	BSO-kPC	PBA	MCaBSO
5	2.139	2.234	2.273	2.361	2.379
10	3.775	3.852	3.864	4.166	4.161
15	5.522	5.561	5.639	5.999	6.023
20	7.502	7.689	7.796	7.894	7.900
25	8.980	9.141	9.263	9.628	9.634
30	10.860	10.933	11.069	11.572	11.540
35	12.077	12.340	12.737	13.205	13.211
40	14.169	14.225	14.591	15.137	15.141
45	15.408	15.814	16.492	16.942	16.955
50	17.197	17.590	18.207	18.758	18.716

B. EXPERIMENTS RESULTS

In this section, we first evaluate the effectiveness using the average and best fitness, and then assess the efficiency using convergence time.

1) EFFECTIVENESS

a: AVERAGE FITNESS EVALUATION

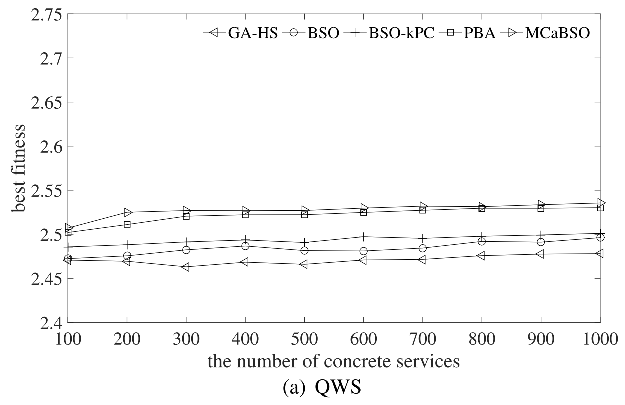
Fig.7 shows the solutions' average fitness values in scenario#1. The values of Fig.7(a) and Fig.7(b) are computed according to the records of QWS and WS-Dream datasets, respectively. Through comparison with the observed data, we can see that the values for PBA and MCaBSO are higher than the values for GA-HS, BSO and BSO-kPC; the values for MCaBSO are slightly higher than those of PBA in most cases. In addition, the overall fitness values increase with the number of concrete services.

Table 3 and Table 4 show the solutions' average fitness values with QWS and WS-Dream datasets in scenario#2. From these tables, we can see that the values for PBA and MCaBSO are bigger than that for GA-HS, BSO and BSO-kPC. The values for PBA are slightly smaller than those of MCaBSO in most cases. As the number of abstract services increases, the values for the five methods increase.

TABLE 4. Average fitness for WS-Dream in scenario#2.

Abstract Services	GA-HS	BSO	BSO-kPC	PBA	MCaBSO
5	2.251	2.321	2.320	2.429	2.419
10	3.005	3.303	3.990	4.145	4.212
15	5.714	5.857	5.904	5.947	6.022
20	6.476	7.092	7.438	7.852	7.894
25	8.235	8.656	8.908	9.661	9.674
30	9.946	10.317	10.922	11.405	11.480
35	11.640	12.661	12.830	12.955	13.103
40	13.475	14.509	14.583	14.803	14.880
45	15.315	16.091	16.306	16.927	16.956
50	16.022	17.825	18.068	18.337	18.414

Through the four sets of experiments, we can conclude that MCaBSO has a better effect in terms of average fitness. Meanwhile, the size of the dataset does not impact the result of the five methods. By adding more quality levels to the selection of services, the potential for the higher fitness values of composite services can be increased at a set of high-quality solutions. Like we described in Section V, MCaBSO utilizes an adaptive dual strategy to adjust the applications of the intra-cluster and inter-cluster operators in different stages. It can effectively exploit the local optima and explore further areas in the search space, so that there are more high-quality combinations. Meanwhile, we can also conclude that by adding concrete services or abstract services to the work-flow of service composition, the average fitness value of composite services can be increased. With the increase of concrete services, there are more web services with better QoS values participation in service composition, so that there are more composite services with higher fitness values. With the increase of abstract services, there are more component services involved in service composition. As a result, the QoS of the composite service with more web services would increase.



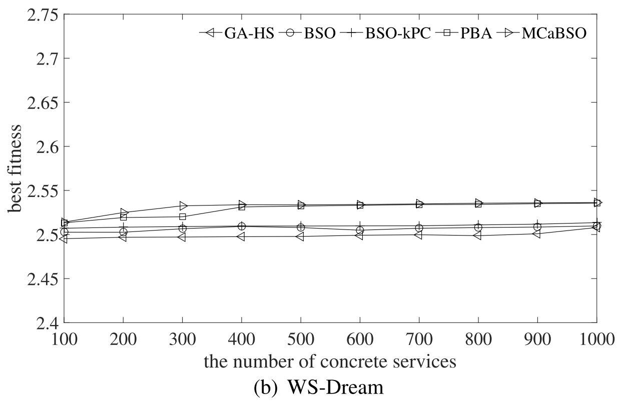


FIGURE 8. Best fitness in secnario#1.

TABLE 5. Best fitness for QWS in scenario#2.

Abstract Services	GA-HS	BSO	BSO-kPC	PBA	MCaBSO
5	2.470	2.491	2.485	2.502	2.507
10	4.075	4.251	4.394	4.411	4.414
15	5.852	6.057	6.155	6.262	6.279
20	7.914	8.059	8.116	8.191	8.208
25	9.566	9.898	9.918	10.086	10.104
30	11.527	11.824	11.939	11.956	11.960
35	13.255	13.536	13.561	13.757	13.753
40	15.069	15.406	15.478	15.675	15.691
45	17.136	17.357	17.436	17.553	17.539
50	19.099	19.189	19.205	19.284	19.286

TABLE 6. Best fitness for WS-Dream in scenario#2.

Abstract Services	GA-HS	BSO	BSO-kPC	PBA	MCaBSO
5	2.495	2.502	2.507	2.513	2.514
10	4.422	4.431	4.432	4.448	4.440
15	6.323	6.339	6.379	6.404	6.420
20	8.025	8.153	8.178	8.194	8.219
25	9.999	10.030	10.042	10.096	10.097
30	11.022	11.871	11.880	11.924	11.932
35	12.821	13.625	13.631	13.629	13.633
40	14.561	15.478	15.481	15.493	15.493
45	16.440	17.364	17.409	17.435	17.434
50	18.220	19.070	19.091	19.145	19.148

b: BEST FITNESS EVALUATION

The experiment evaluations of the best fitness in scenario#1 are shown in Fig.8, where Fig.8(a) displays the results for QWS dataset and Fig.8(b) shows the results for WS-Dream dataset. From the two figures, we see that MCaBSO and PBA have higher values than GA-HS, BSO and BSO-kPC, and the values of MCaBSO are slightly higher than those of PBA in most cases.

The experiment results in evaluating the best fitness in scenario#2 are reported in Table 5 and Table 6. Table 5 shows the best fitness values, which are computed from the QWS dataset and Table 6 reports the best fitness values for WS-Dream dataset. By comparing the data for each row, the values for MCaBSO and PBA are bigger than that of GA-HS, BSO and BSO-kPC; the values for PBA are slightly smaller than that for MCaBSO in most cases. By comparing the data for each column, we can see that the values increase with the number of abstract services.

From the experiments, we can conclude that MCaBSO has better effectiveness in terms of best fitness values. Like the mechanism we used in Section V, TWSVM clusters similar individuals according to the natural organization of individuals. Multiple clusters interact with their feature information of QoS and exploit them to compose better individuals. To void crowd gathering behavior, the mutation with a disturbance is added to improve the effectiveness of exploration. Meanwhile, we can also conclude that by adding concrete services or abstract services to the service composition, the optimality can be further improved. With the increase of concrete services, the optimal solution would be explored by the active exploration of MCaBSO. With the rise of abstract services, the fitness value would include more component services so that the fitness value increases.

2) EFFICIENCY

The experimental results of evaluating the five methods' efficiency in scenario#1 and scenario#2 are shown in Fig.9 and Fig.10. The first two sets of experiments are conducted in scenario#1. Their convergence times for the QWS and WS-Dream datasets are shown in Fig.9(a) and Fig.9(b), respectively. In Fig.9, lines with left triangle markers, circle markers, plus markers, square markers and right triangle markers, depict the convergence time of GA-HS and BSO, BSO-kPC, PBA, MCaBSO, respectively. From the comparison of the convergence times, we can see that the values of MCaBSO are smaller than other algorithms. As the increase of concrete services, the convergence times of the five methods also increase.

The latter two sets of experiments are conducted in scenario#2. Fig.10(a) and Fig.10(b) show the convergence time for QWS and WS-Dream datasets, respectively. In the two figures, lines with left triangle markers, circle markers,

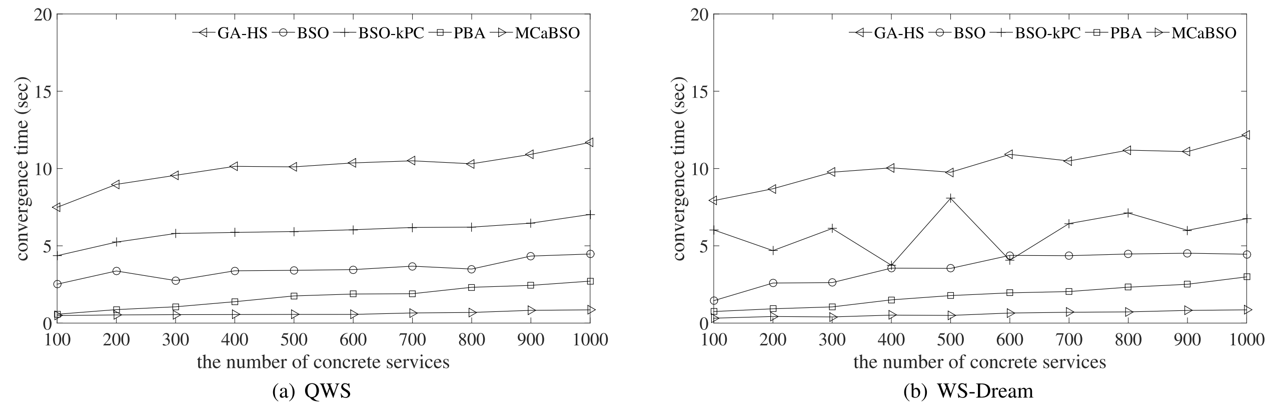


FIGURE 9. Convergence time in scenario#1.

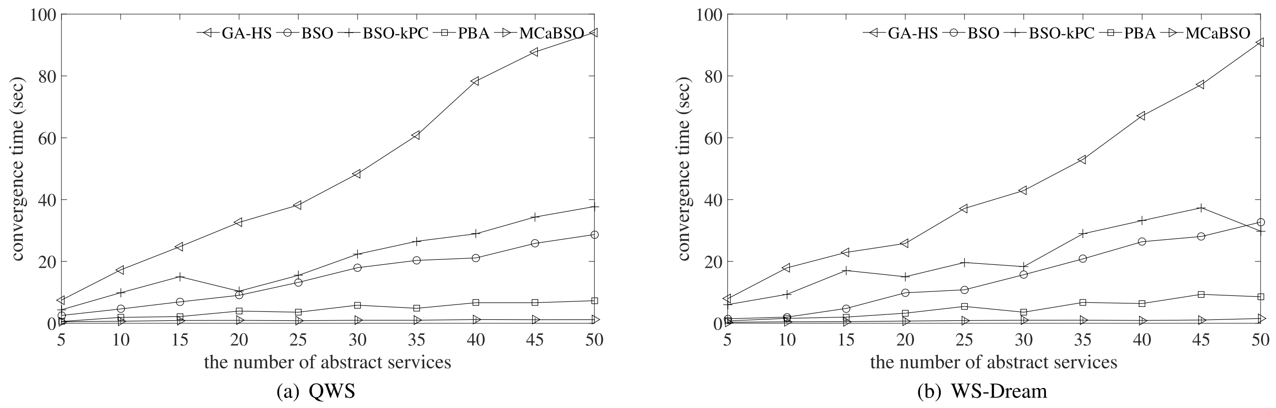


FIGURE 10. Convergence time in scenario#2.

plus markers, square markers and right triangle markers, depict the convergence time of GA-HS and BSO, BSO-kPC, PBA, MCaBSO, respectively. The values of MCaBSO are smaller than those of other methods. We also notice that the values increase with the growth of abstract services.

Through the four sets of experiments, we can conclude that by using the problem reduction and utilizing information of services for guiding the exploration of solutions space, the convergence time of finding the optimal solution can be reduced. As we represented in Section V, TWSVM divides the optimization problem into multiple subsets optimization problems and downsizes the scale. Then various clusters with different QoS levels are collaboratively finding the optimal solution by exploring their subsets and exchanging the optimal information among subsets. Therefore, the size for search space can be reduced and the solutions with higher fitness values can be refined in the same area. Meanwhile, we can also conclude that the convergence time of finding the optimal composite service would grow with the number of concrete services and abstract services. The growth in the number of concrete services and abstract services expands the size of the search space. As a result, more time is required to find more composite services and compare fitness values. Since the number of abstract services has a greater impact on the search space than the number of concrete services, the growth rate in the time of abstract services is higher than that in the convergence time of concrete services.

C. SUMMARY OF RESULTS

From the experimental results, the evaluation values may be different for two different datasets, but the conclusions are similar. MCaBSO and PBA are distinctly better than other algorithms, as well as MCaBSO is slightly better than PBA in the aspect of average fitness value and optimal fitness value. In terms of convergence time, MCaBSO is significantly better than other algorithms. It is verified that MCaBSO outperforms different algorithms in terms of effectiveness and efficiency. With the increase in the number of abstract services and concrete services, MCaBSO would spend less convergence time to achieve optimal or close-to-optimal results. It is obvious that MCaBSO is more scalable than other methods.

VII. CONCLUSION AND FUTURE WORK

In a service-oriented computing environment, service composition provides a means to develop complex applications and satisfy the user's personalized requirements. With the fast growth of web services, each user requirement may be supported by a set of services with similar functionality but different QoS. In this paper, a hybrid method is proposed to solve the QoS-aware service composition. This method uses BSO as the underlying search scheme to combine the search space reduction and the exploration of the reduced search space. Also, TWSVM is used to construct nonparallel planes to enable the division of the search space, and an adaptive dual strategy is provided for the adjustment of using intra-cluster and inter-cluster evolutionary operators in different stages. The selection policy emerges into the selection operator for obtaining elites. It handles QoS-aware service composition optimization effectively and efficiently. The proposed method is evaluated on two data sets, different evaluation indicators, varying requirements through the number of concrete services, or the number of abstract services. The results show the hybrid method handles the optimization of service composition effectively and efficiently by comparing it with other methods. In future work, we plan to investigate multi-objective optimization further. More quality metrics (e.g., reliability, availability) will be discussed. This is critical because the optimal composite service could not be reached if the weights of evaluating user's preferences are fuzzy and the number of QoS attributes increases. Furthermore, the effectiveness of other cluster techniques and selection policies will be analyzed.

ACKNOWLEDGMENT

The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency.

REFERENCES

[1] M. Alrifai, D. Skoutas, and T. Risse, "Selecting skyline services for QoS-based Web service composition," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 11–20.

[2] L. Wang, "Architecture-based reliability-sensitive criticality measure for fault-tolerance cloud applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 11, pp. 2408–2421, Nov. 2019.

[3] *Amazon Elastic Compute Cloud (Amazon EC2)*. Accessed: Mar. 10, 2020. [Online]. Available: https://aws.amazon.com/ec2/

[4] J. Li, Y. Yan, and D. Lemire, "Scaling up Web service composition with the skyline operator," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2016, pp. 147–154.

[5] T. H. Tan, M. Chen, J. Sun, Y. Liu, É. André, Y. Xue, and J. S. Dong, "Optimizing selection of competing services with probabilistic hierarchical refinement," in *Proc. 38th Int. Conf. Softw. Eng. (ICSE)*, 2016, pp. 85–95.

[6] P. Rodriguez-Mier, M. Mucientes, and M. Lama, "Hybrid optimization algorithm for large-scale QoS-aware service composition," *IEEE Trans. Services Comput.*, vol. 10, no. 4, pp. 547–559, Jul. 2017.

[7] S.-L. Fan, K.-Y. Peng, and Y.-B. Yang, "Large-scale QoS-aware service composition integrating chained dynamic programming and hybrid pruning," in *Proc. Int. Conf. Web Services*. Cham, Switzerland: Springer, 2018, pp. 196–211.

[8] X. Wang, X. Xu, Q. Z. Sheng, Z. Wang, and L. Yao, "Novel artificial bee colony algorithms for QoS-aware service selection," *IEEE Trans. Services Comput.*, vol. 12, no. 2, pp. 247–261, Mar. 2019.

[9] H. Wang, G. Huang, and Q. Yu, "Automatic hierarchical reinforcement learning for efficient large-scale service composition," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2016, pp. 57–64.

[10] J. Zhou and X. Yao, "Multi-population parallel self-adaptive differential artificial bee colony algorithm with application in large-scale service composition for cloud manufacturing," *Appl. Soft Comput.*, vol. 56, pp. 379–397, Jul. 2017.

[11] M. S. Hossain, M. Moniruzzaman, G. Muhammad, A. Ghoneim, and A. Alamri, "Big data-driven service composition using parallel clustered particle swarm optimization in mobile environment," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 806–817, Sep. 2016.

[12] Y. Shi, "Brain storm optimization algorithm," in *Proc. Int. Conf. Swarm Intell.* Berlin, Germany: Springer, 2011, pp. 303–309.

[13] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, May 2007.

[14] Z. Wang, Y.-H. Shao, L. Bai, and N.-Y. Deng, "Twin support vector machine for clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2583–2588, Oct. 2015.

[15] S. Cheng, Y. Shi, Q. Qin, T. O. Ting, and R. Bai, "Maintaining population diversity in brain storm optimization algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 3230–3237.

[16] Y. Yang, Y. Shi, and S. Xia, "Advanced discussion mechanism-based brain storm optimization algorithm," *Soft Comput.*, vol. 19, no. 10, pp. 2997–3007, Oct. 2015.

[17] H. Peng, C. Deng, and Z. Wu, "SPBSO: Self-adaptive brain storm optimization algorithm with pbest guided step-size," *J. Intell. Fuzzy Syst.*, vol. 36, no. 6, pp. 5423–5434, Jun. 2019.

[18] F. Pourpanah, C. P. Lim, X. Wang, C. J. Tan, M. Seera, and Y. Shi, "A hybrid model of fuzzy min–max and brain storm optimization for feature selection and data classification," *Neurocomputing*, vol. 333, pp. 440–451, Mar. 2019.

[19] H. Duan, S. Li, and Y. Shi, "Predator–prey brain storm optimization for DC brushless motor," *IEEE Trans. Magn.*, vol. 49, no. 10, pp. 5336–5340, Oct. 2013.

[20] A. Aldhafeeri, "Brain storm optimization for electromagnetic applications," Ph.D. dissertation, University of California, Los Angeles, Los Angeles, CA, USA, 2018.

[21] X. Ma, Y. Jin, and Q. Dong, "A generalized dynamic fuzzy neural network based on singular spectrum analysis optimized by brain storm optimization for short-term wind speed forecasting," *Appl. Soft Comput.*, vol. 54, pp. 296–312, May 2017.

[22] H. Qiu and H. Duan, "Receding horizon control for multiple UAV formation flight based on modified brain storm optimization," *Nonlinear Dyn.*, vol. 78, no. 3, pp. 1973–1988, Nov. 2014.

[23] M. Sato and Y. Fukuyama, "Total optimization of smart city by global-best brain storm optimization," in *Proc. Genetic Evol. Comput. Conf. Companion (GECCO)*, 2018, pp. 304–305.

[24] D. Palanikkumar and S. Priya, "Brain storm optimization graph theory (BSOGT) and energy resource aware virtual network mapping (ERVNM) for medical image system in cloud," *J. Med. Syst.*, vol. 43, no. 2, p. 37, Feb. 2019.

[25] J.-H. Hao, J.-Q. Li, Y. Du, M.-X. Song, P. Duan, and Y.-Y. Zhang, "Solving distributed hybrid flowshop scheduling problems by a hybrid brain storm optimization algorithm," *IEEE Access*, vol. 7, pp. 66879–66894, 2019.

[26] D. Li, D. Ye, N. Gao, and S. Wang, "Service selection with QoS correlations in distributed service-based systems," *IEEE Access*, vol. 7, pp. 88718–88732, 2019.

[27] N. Zorba, A. I. Pérez-Neira, A. Foglar, and C. Verikoukis, "Cross layer QoS guarantees in multiuser WLAN systems," *Wireless Pers. Commun.*, vol. 51, no. 3, pp. 549–563, Nov. 2009.

[28] X. Sun, S. Wang, Y. Xia, and W. Zheng, "Predictive-trend-aware composition of Web services with time-varying quality-of-service," *IEEE Access*, vol. 8, pp. 1910–1921, 2020.

[29] D. Ardagna and B. Pernici, "Adaptive service composition in flexible processes," *IEEE Trans. Softw. Eng.*, vol. 33, no. 6, pp. 369–384, Jun. 2007.

[30] M. E. Khanouche, F. Attal, Y. Amirat, A. Chibani, and M. Kerkar, "Clustering-based and QoS-aware services composition algorithm for ambient intelligence," *Inf. Sci.*, vol. 482, pp. 419–439, May 2019.

[31] S. K. Gavvala, C. Jatoth, G. R. Gangadharan, and R. Buyya, "QoS-aware cloud service composition using eagle strategy," *Future Gener. Comput. Syst.*, vol. 90, pp. 273–290, Jan. 2019.

[32] H. Alayed, F. Dahan, T. Alfakih, H. Mathkour, and M. Arafah, "Enhancement of ant colony optimization for QoS-aware Web service selection," *IEEE Access*, vol. 7, pp. 97041–97051, 2019.

[33] S. Chattopadhyay and A. Banerjee, "QoS constrained large scale Web service composition using abstraction refinement," *IEEE Trans. Services Comput.*, to be published.

[34] Z. Wang, B. Cheng, W. Zhang, and J. Chen, "Q-graphplan: QoS-aware automatic service composition with the extended planning graph," *IEEE Access*, vol. 8, pp. 8314–8323, 2020.

[35] S. Deng, L. Huang, W. Tan, and Z. Wu, "Top-*k* automatic service composition: A parallel method for large-scale service sets," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 891–905, Mar. 2014.

[36] H. Wang, S. Peng, and Q. Yu, "A parallel refined probabilistic approach for QoS-aware service composition," *Future Gener. Comput. Syst.*, vol. 98, pp. 609–626, Sep. 2019.

[37] K. P. Yoon and C.-L. Hwang, *Multiple Attribute Decision Making: An Introduction*, vol. 104. Newbury Park, CA, USA: Sage, 1995.

[38] H. Duan and C. Li, "Quantum-behaved brain storm optimization approach to solving Loney's solenoid problem," *IEEE Trans. Magn.*, vol. 51, no. 1, Jan. 2014, Art. no. 7000307.

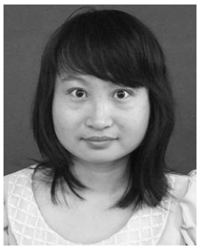
[39] W. Zhen, C. Jin, and Q. Ming, "Non-parallel planes support vector machine for multi-class classification," in *Proc. Int. Conf. Logistics Syst. Intell. Manage. (ICLSIM)*, vol. 1, Jan. 2010, pp. 581–585.

[40] E. Al-Masri and Q. H. Mahmoud, "Discovering the best Web service," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, 2007, pp. 1257–1258.

[41] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed QoS evaluation for real-world Web services," in *Proc. IEEE Int. Conf. Web Services*, Jul. 2010, pp. 83–90.

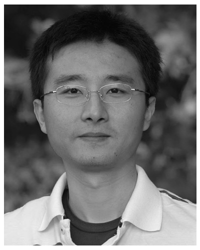
[42] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *J. Global Optim.*, vol. 16, no. 1, pp. 23–32, Jan. 2000.

[43] A. E. Yilmaz and P. Karagoz, "Improved genetic algorithm based approach for QoS aware Web service composition," in *Proc. IEEE Int. Conf. Web Services*, Jun. 2014, pp. 463–470.



SHUNSHUN PENG is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Southeast University, China. Her publications have mainly appeared in international conferences and journals, i.e., ICWS and FGCS. Her research interests include service computing and data mining.

HONGBING WANG (Member, IEEE) received the Ph.D. degree in computer science from Nanjing University, China. He is currently a Professor from the School of Computer Science and Engineering, Southeast University, China. He published more than 50 refereed articles in international conferences and journals, e.g., the *Journal of Web Semantics*, TSE, TAAS, JSS, TSC, ICSOC, ICWS, SCC, CIKM, ICTAI, and WI. His research interests include service computing, cloud computing, and software engineering.



QI YU (Member, IEEE) received the Ph.D. degree in computer science from the Virginia Polytechnic Institute and State University (Virginia Tech). He is currently an Associate Professor with the College of Computing and Information Sciences, Rochester Institute of Technology. His current research interests are in the areas of service computing, databases, and data mining. His publications have mainly appeared in well-known journals (e.g., *The VLDB Journal*, the *ACM Transactions on the Web*, *World Wide Web Journal*, and the IEEE TRANSACTIONS ON SERVICES COMPUTING) and conference proceedings (e.g., ICSOC and ICWS).