

Research Article

Multi-Objective Service Composition Using Enhanced Multi-Objective Differential Evolution Algorithm

Shunshun Peng ¹ and Taolin Guo ²

¹College of Computer and Information Sciences, Chongqing Normal University, Chongqing 401331, China

²College of Computer Science, Chongqing University, Chongqing 400044, China

Correspondence should be addressed to Taolin Guo; tguo@cqu.edu.cn

Received 8 October 2022; Revised 13 February 2023; Accepted 16 February 2023; Published 4 March 2023

Academic Editor: Rodolfo E. Haber

Copyright © 2023 Shunshun Peng and Taolin Guo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, the optimization of multi-objective service composition in distributed systems has become an important issue. Existing work makes a smaller set of Pareto-optimal solutions to represent the Pareto Front (PF). However, they do not support complex mapping of the Pareto-optimal solutions to quality of service (QoS) objective space, thus having limitations in providing a representative set of solutions. We propose an enhanced multi-objective differential evolution algorithm to seek a representative set of solutions with good proximity and distributivity. Specially, we propose a dual strategy to adjust the usage of different creation operators, to maintain the evolutionary pressure toward the true PF. Then, we propose a reference vector neighbor search to have a fine-grained search. The proposed approach has been tested on a real-world dataset that locates a representative set of solutions with proximity and distributivity.

1. Introduction

Service composition became popular after introducing service-oriented architecture (SOA), as it allows complex and distributed software systems to be composed of web services through open standards. QoS attributes [1] (e.g., reliability or throughput) provide the quality criteria for selecting and composing web services, thus establishing QoS-aware service composition (QOSC). Since QoS requirements usually involve multiple conflicting objectives, QOSC is a multi-objective optimization problem (MOP) to find a set of Pareto-optimal solutions.

Existing work [2–4] has explored multi-objective evolutionary algorithm (MOEA) that allows a set of feasible solutions to approximate the Pareto-optimal set, based on analyzing a set of non-dominated solutions after one run and maintaining a good solution diversity during the search [5, 6]. However, this approach includes elitist preservation in the selection strategy. Therefore there may be a lack of evolutionary pressure to explore optimal solutions, especially as the number of service requests increases. Second, this approach

does not explicitly consider fine-grained search. This can lead to overlapped or unstructured searches, resulting in uneven distribution of Pareto-optimal solutions.

In this work, we tackle the issue by proposing an enhanced multi-objective differential evolution (EnMODE) algorithm for searching for a representative set of feasible solutions to approximate the Pareto-optimal solutions in terms of proximity and distributivity. Our main contributions are summarized as follows:

- (1) Sufficient evolutionary pressure - We propose a dual strategy to adjust the usage of “rand/2/bin” and “current-to-best/1/bin” as the iteration evolves. “rand/2/bin” expands the evolution ability by performing new exploration around two different solutions. At the same time “current-to-best/1/bin” improves the evolution robustness by performing guided exploration around the current best. The dynamic execution of “rand/2/bin” and “current-to-best/1/bin” provides sufficient evolutionary pressure as the population evolves.

- (2) Fine-grained search - The reference vectors are used as neighborhood axis to divide MOP, and accumulate the non-dominated compositions around the nearest reference vector and the dominated compositions around the essential non-dominated compositions. The reference vector neighbor search simplifies MOP by systematically breaking it down into similar sub-problems and directing the search to the Pareto-optimal set.

We have evaluated the enhanced multi-objective differential evolution algorithm on a real-world dataset, which shows that the proposed method has better proximity and distributivity than the baselines. The rest of the paper is organized as follows. In Section 2, we compare it with related work. Section 3 introduces the multi-objective service composition model. We outline the proposed approach for the MOP in Section 4. Section 5 evaluates the proposed method in terms of proximity degree and uniformity. Finally, Section 6 concludes the paper and outlines future work.

2. Related Work

This section outlines the related work on MOSC. Section 2.1 focuses on Pareto-based multi-objective service composition (PMOSC), which finds a set of Pareto-optimal solutions based on the hypothesis that users cannot accurately pre-define weights or priorities for multidimensional objectives. Section 2.2 presents the utility-based multi-objective service composition (UMOSC), which computes the utility value of a solution based on the hypothesis that the weights or priorities can be accurately specified. The summary of the related work is presented in Section 2.3.

2.1. Pareto-Based Multi-Objective Service Composition. The intuitive method is to explore all Pareto-optimal solutions exhaustively. Since the Pareto-optimal set may include all possible solutions that exponentially grow in the sizes of service requests, the optimization cost of such a method would be prohibitive.

To solve this problem, Guo et al. [7] proposed a computationally efficient dropout neural network as a computationally scalable alternative of the Gaussian process model for assisting the solution of expensive high-dimensional multi-objective and many-objective expensive optimization problems. Li et al. [8] built the energy-efficient job-shop scheduling problem to a many-objective model with five objectives, i.e., makespan, total tardiness, total idle time, total worker cost, and total energy. They adopted a novel fitness evaluation mechanism based on fuzzy correlation entropy to solve this many-objective optimization problem. Cruz et al. [9] proposed an evolutionary algorithm-based search strategy for choosing an efficient design of an ensemble of Convolutional Neural Networks (CNNs), which includes not only the networks architecture but also the voting policy. During the running of the search strategy, not only the combination of CNNs with different architectures is taken into consideration, but also the most suitable policy used by the ensemble for generating the unified response.

Zhou et al. [3] proposed a multi-population differential artificial bee colony optimizer for PMOSC. The optimization problem is divided into several sub-problems to reduce the search scale. Different search behaviors are considered in the artificial bee colony algorithm to select the solutions set toward the Pareto-optimal set. The work of [10] integrated hyper-heuristics with genetic programming to solve the multi-objective dynamic service composition optimization. A set of Pareto-optimal solutions are provided to satisfy varied preferences. Wang et al. [11] proposed an improved whale optimization algorithm to divide the population into several populations. A pareto strategy is presented to improve the optimization. Yang et al. [12] adopted a multi-objective immune algorithm to implement PMOSC. The global ranking is incorporated into the evolution of multiple populations to obtain better generations.

Chen et al. [13] proposed an objective space partition-based adaptive multi-objective evolutionary algorithm to maintain diversity during strength convergence. The proposed approach defines the forward population distance as a metric to dynamically identify efficient subspaces and adaptively allocate computational resources to each subspace. In [14], an enhanced decomposition-based evolutionary many-objective optimization algorithm is proposed to solve irregular many-objective optimization problems. The local search is performed on external archives to alleviate the adverse effects of inappropriate weight vectors and strengthen the performance. Dai et al. [15] proposed a problem-specific multi-objective evolutionary algorithm where a decomposition scheme decomposes PMOSC into multiple scalar sub-problems. The evolutionary operators search Pareto solutions in terms of maximizing the service quality and minimizing the overhead. Seada and Deb [16] developed a unified evolutionary optimization algorithm U-NSGA-III to solve mono-, multi-, and many-objective optimization problems. The ability of U-NSGA-III to solve different types of problems equally efficiently and sometimes better, with the added flexibility brought in through population size control, remains a hallmark achievement. Dhiman et al. [17] proposed a novel hybrid many-objective evolutionary algorithm named Reference Vector Guided Evolutionary Algorithm (H-RVEA). It decomposed the optimization problem into several sub-problems by reference vectors, and used an adaptation strategy to adjust the reference vector distribution.

Lin et al. [18] proposed an adaptive immune-inspired multi-objective algorithm. This method embeds three differential evolution (DE) strategies with distinct features into multi-objective immune algorithms. At each generation, one of them is adaptively selected to be used based on the current search stage. This adaptive DE strategy selection effectively cooperates with three DE strategies, significantly improving search capability and population diversity. Kumar et al. [19] proposed a differential evolution and sine cosine algorithm-based new hybrid optimization method. This method adapted multi-objective versions of evolutionary optimization-based methods to mine the reduced high-quality numerical association rules automatically. Altay and Alatas [20] proposed an enhanced version of the multi-

operators variant differential evolution model, named ESHADE. ESHADE utilized various mutation strategies, and an exponential population size reduction (EPSR) technique to reduce the population size for the next iteration. Besides, ESHADE employed a version of the univariate sampling method in later iterations to balance exploitative and explorative searches. We can conclude that the multi-operator variant [18–20] implemented the task of enhancing the diversity of the candidate solutions. The difference is that the execution probability for the multi-operator variant is not a quantitative benchmark that accurately reflects the current search stage. At the same time, we calculate the probability as the search continues. This difference leads to problems because the multi-operator variant has difficulty tackling MOPs with different characteristics, and our proposed method solves these problems.

2.2. Utility-Based Multi-Objective Service Composition. Different approaches are developed to find the composition with the best utility, such as graph search [21, 22], evolutionary algorithms [23, 24], and so on.

Rodríguez-Mier et al. [25] used a *Service Match Graph* to represent all matches between the relevant services. On this basis, they proposed a hybrid local/global search to find the optimal solution. Siebert et al. [26] transformed the service composition problem into the subgraph isomorphism problem. A message-efficient localized algorithm is proposed to compose the component services according to the information from the collaboration candidate. There are also existing works on evolutionary algorithms for finding the optimal solution. For example, Hossain et al. [27] extended the particle swarm optimization algorithms to improve global and local optimization. The particles search the service space with guidance from extreme individual value and population extreme value. Martín et al. [28] proposed an ant colony optimization algorithm, in which a set of ants find the shortest path according to the pheromone mechanism. Some works used machine learning technologies to find the optimal solution. Wang et al. [29] integrated reinforcement learning with multi-agent techniques for finding the optimal solution. Game theory and a fictitious play process are combined to help improve performance. Peng et al. [30] and Wang et al. [31] used a restricted Boltzmann machine to learn the probability information of the global optimization contribution of concrete service. The information helps guide the search for solutions. Palade and Clarke [32] adopted collaborative agent communities to approximate the optimal solution.

2.3. Summary. The existing work for UMOsc can find an optimal or near-optimal solution effectively by maximizing or minimizing the utility value, which is computed under the basis of specifying weights. However, it is not easy to determine weights in practice. The reason might be that the information on user preference for multiple attributes is lost. Even if they know the user preference, it is hard to provide accurate quantitative values.

The work for PMOSC finds a set of Pareto-optimal solutions under the assumption of unknown weights, but the performance, such as proximity and distributivity, needs to be improved. In this paper, we focus on the work for PMOSC, and provide a hybrid approach to search for a smaller set of solutions with proximity and distributivity.

3. Multi-Objective Service Composition Model

3.1. QoS Vectors

Definition 1. (QoS vector) Assuming a web service ws has M attributes, the quality of ws is described by M attributes that are considered an M -dimensional QoS vector. Thus, the QoS vector for ws is defined as $Q(ws) = \{a_1(ws), a_2(ws), \dots, a_M(ws)\}$, where $a_m(ws)$ represents the m_{th} QoS attribute value of ws (for $1 \leq m \leq M$).

Definition 2. (QoS vector for a composition) A composition is represented as $cs = (ws_1, ws_2, \dots, ws_n)$, where $ws_i (1 \leq i \leq n)$ is the concrete service of specifying the instantiation of the i_{th} abstract service. The QoS vector for cs is defined as $Q(cs) = \{f_1(cs), f_2(cs), \dots, f_M(cs)\}$, where $f_m(cs)$ is the aggregation value of the m_{th} QoS attribute for all concrete services in cs .

As shown in Table 1, the aggregation value is computed based on the aggregation function. Other QoS attributes share similar aggregation functions, e.g., the cost computation in the case of sequential execution has a similar summation aggregation function.

3.2. Pareto-Optimality

Definition 3. (Pareto-dominance) Given two compositions cs_a and cs_b , their QoS vectors are denoted by $Q(cs_a) = \{f_1(cs_a), f_2(cs_a), \dots, f_M(cs_a)\}$ and $Q(cs_b) = \{f_1(cs_b), f_2(cs_b), \dots, f_M(cs_b)\}$. cs_b is said to Pareto-dominate cs_a if i) for every attribute cs_b has a better QoS value than cs_a or equivalent value like cs_a , and ii) for some attributes, cs_b has better QoS values than cs_a , i.e.

$$\begin{aligned} \forall m \in \{1, \dots, M\}, f_m(cs_a) \leq f_m(cs_b) \wedge \\ \exists m \in \{1, \dots, M\}, f_m(cs_a) < f_m(cs_b). \end{aligned} \quad (1)$$

For brevity, the relation that cs_b dominates cs_a is denoted by $cs_a < cs_b$. Through the notion of Pareto-dominance, we can also determine that cs_b and cs_a are non-dominated by each other if neither $cs_a < cs_b$ nor $cs_b < cs_a$.

Definition 4. (Pareto-optimality) Given a set of compositions CS , a composition $cs \in CS$ is a Pareto-optimal solution if it is feasible and not strictly dominated by any other feasible composition cs' , i.e.

$$\nexists cs' \in CS: cs' > cs. \quad (2)$$

Consider a set of feasible compositions, and we have the relations $cs_4 > cs_1, cs_4 > cs_2, cs_4 > cs_3, cs_6 > cs_5, cs_6 > cs_7$, and

TABLE 1: Examples of QoS aggregation functions.

Attributes	Sequential	Parallel	Loop	Conditional
Response time	$\sum_{i=1}^n q(ws_i)$	$\max_{i=1}^n q(ws_i)$	$k * q(ws_i)$	$\max_{i=1}^n q(ws_i)$
Throughput	$\min_{i=1}^n q(ws_i)$	$\sum_{i=1}^n q(ws_i)$	$\min_{i=1}^n q(ws_i)$	$\max_{i=1}^n q(ws_i)$

$cs_6 > cs_8$. Since other compositions do not dominate cs_4 and cs_6 , they are Pareto-optimal.

3.3. Problem Statement. The MOP of service composition can be defined as follows:

$$\begin{aligned} \max Q(cs) &= (f_1(cs), f_2(cs), \dots, f_M(cs)), \\ &\text{subject to } cs \in CS, \end{aligned} \quad (3)$$

where cs represents a composition $(ws_1, ws_2, \dots, ws_n)$, CS is the set of composite services.

Considering multiple QoS attributes, service composition optimization is regarded as MOP. Due to conflicting objectives and unavailable preferences, finding a solution with the best values for all objectives is complex. An intuitive method to address this problem is to explore all Pareto-optimal solutions. However, the solution space size grows exponentially as service requests increase. Finding all Pareto-optimal solutions will cost a lot. Therefore, it is more desirable to approximate the Pareto-optimal set to allow runtime multi-objective service composition. This work aims to solve multi-objective service composition by seeking a representative set of solutions with good proximity and distributivity in QoS objective space.

4. The EnMODE for Multi-Objective Service Composition

4.1. Initialization. The initialization of the proposed approach is conducted from two aspects: the population and the reference vectors.

An individual corresponds to a composite service composed of several concrete services for the population. The concrete service is randomly chosen from the service candidates. A unique identifier $id(ws)$ is used to identify the concrete service ws . After identifying the concrete service, an individual cs is represented as $cs = (id^1(ws), id^2(ws), \dots, id^D(ws))$, where D is the number of abstract services. The QoS vector of cs is represented as $Q(cs) = \{f_1(cs), f_2(cs), \dots, f_M(cs)\}$, where M is the size of the objectives.

For the initialization of the reference vectors, the key steps are listed. First, the reference point is generated by sampling points on a hyperplane. Then the reference points are mapped on the PF to generate the reference vectors. A reference vector is a vector that starts from the origin point in the objective space and ends in the reference point. Let H be the parameter that controls the division on the objective axis, a reference vector $\lambda_i = (\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{iM})$ is generated by selecting λ_{im} from $\{0/H, 1/H, \dots, H/H\}$ and satisfying $\sum_{m=1}^M \lambda_{im} = 1$. Usually, the number of reference vectors

equals the population size N . The initial reference vector is stored in λ^0 . Therefore, the initial reference vectors are represented as $\lambda^0 = (\lambda_1^0, \dots, \lambda_N^0)$.

4.2. Offspring Creation with Dual Strategy. According to references [33], the state of the search space varies with the evolutionary process. At the early stage, the available information (e.g., the individuals with better QoS) about the search is limited. The available information about the search is accumulated with the increase of the iterations. Entering the latter stage, many individuals might be close to the true PF, so it is less likely to find better individuals even if taking longer. Such a change would leave the population without evolutionary pressure. To solve this problem, a dual strategy is needed to tweak the use of creation operators to provide sufficient evolutionary pressure in the evolutionary stage.

First, two creation operators, namely, “rand/2/bin” and “current-to-best/1/bin”, are selected to manipulate individuals. The “rand/2/bin” shows better disturbance and creates offspring based on different individuals. These characteristics make it capable of expanding its evolution ability. The “current-to-best/1/bin” creates the offspring by searching around the current best, which improves the evolutionary robustness. These two creation operators regard the abstract service as the operational dimension to generate a new individual. The new individual $cs' = (id^1(ws'), \dots, id^D(ws'))$ is produced as follows.

(1) rand/2/bin

$$id_i^d(ws') = \begin{cases} id_{r_1}^d(ws) + F_1 \times (id_{r_2}^d(ws) - id_{r_s}^d(ws)) \\ + F_2 \times (id_{r_4}^d(ws) - id_{r_s}^d(ws)), \\ \text{if } \text{rand}(0, 1) < 0.9 \text{ or } d = d_{\text{rand}} \\ id_i^d(ws), \text{ otherwise.} \end{cases} \quad (4)$$

(2) current-to-best/1/bin

$$id_i^d(ws') = \begin{cases} id_i^d(ws) + F_3 \times (id_{\text{best}}^d(ws) - id_i^d(ws)) \\ + F_4 \times (id_{r_1}^d(ws) - id_{r_2}^d(ws)), \\ \text{if } \text{rand}(0, 1) < 0.1 \text{ or } d = d_{\text{rand}} \\ id_i^d(ws), \text{ otherwise,} \end{cases} \quad (5)$$

where r_1, r_2, r_3, r_4 , and r_s are the indices of different random individuals in the current population; $best$ is the index of the best individual; d_{rand} is the random index within the range $[1, \dots, D]$; F_1, F_2, F_3 and F_4 control the proportion of different individuals.

Due to the finiteness of the service candidates, the identifier of the concrete service might exceed its limit. Therefore, the identifier needs to be reset, which is conducted according to the following formula.

$$\text{id}_i^d(ws)'' = \text{id}_i^d(ws)' + \sigma_i \times (\text{id}_i^d(ws)^{\max} - \text{id}_i^d(ws)^{\min}), \quad (6)$$

with

$$\sigma_i = \begin{cases} (2 \times \text{rand}(0, 1))^{1/20} - 1, & \text{if } \text{rand}(0, 1) < 0.5, \\ 1 - (2 - 2 \times \text{rand}(0, 1))^{1/20}, & \text{otherwise,} \end{cases} \quad (7)$$

where $\text{id}_i^d(ws)^{\max}$ and $\text{id}_i^d(ws)^{\min}$ represent the upper and low bounds of the identifier, respectively.

Second, these two operators are adjusted by a dual strategy, which controls the execution probability of “rand/2/bin” and “current-to-best/1/bin” in the evolutionary stage. There is no clear division between the various evolutionary stages throughout the evolution process. Therefore, we define the ratio of the current iteration to the total iteration to distinguish between the different evolutionary stages. On account of the ratio, we represent their probabilities as p_{rand} and p_{best} , and compute them as follows.

$$p_{\text{rand}} = \frac{1}{1 + \exp(10 \times ((i/\text{iter}_{\max}) - 0.5))}, \quad (8)$$

$$p_{\text{best}} = 1 - p_{\text{rand}},$$

where i is the number of the current iteration and iter_{\max} is the number of total iteration. To exhibit the whole dynamic changes of p_{rand} and p_{best} more clearly, their tendencies are illustrated in Figure 1. At the early stage, the “rand/2/bin” is run with a high probability of exploring more high-quality individuals. As the evolution continues, the execution probability of the “rand/2/bin” dynamically reduces. In contrast, the execution probability of the “current-to-best/1/bin” dynamically increases. The exploration of the “rand/2/bin” and the exploitation of the “current-to-best/1/bin” are used to speed up convergence and prevent premature convergence. Entering the latter stage, the “current-to-best/1/bin” is preferred to exploit the local information.

4.3. Reference Vector Neighbor Search. This search first unfolds the two-stage clustering to implement a fine-grained search under the guidance of the reference vectors and the elites in the non-dominated individuals, and then carries out the selection of the solutions set towards the true PF.

4.3.1. Two-Stage Clustering. In the two-stage clustering, the first-stage clustering uses the reference vector as the anchor to gather the non-dominated individuals. The second-stage clustering groups the dominated individuals under the direction of the elites in the first cluster.

For the first-stage clustering, we first search the non-dominated individuals from the union of the offspring and the current population. Then, the non-dominated

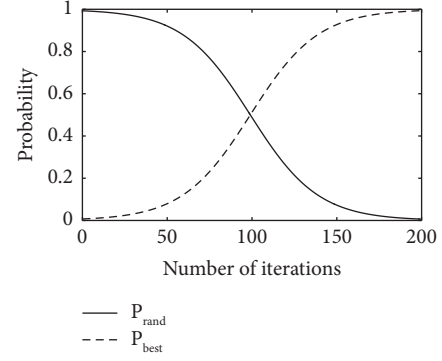


FIGURE 1: The tendencies of p_{rand} and p_{best} .

individuals are clustered by computing their closeness degree with the reference vectors. The closeness degree can be measured by the perpendicular distance from the individual cs to a reference vector λ . The perpendicular distance is computed as follows.

$$d(Q(cs), \lambda) = \sqrt{\|Q(cs)\|_2^2 - d_1^2}, \quad (9)$$

where d_1 represents the distance along λ , which is computed by $d_1 = Q(cs)\lambda^T$, and $\|\cdot\|_2$ represents the l_2 norm of the vector. Each non-dominated individual can be attached to the nearest reference vector by comparing the distance values. Reference vectors attached by non-dominated individuals are labeled as active, while reference vectors without attached individuals are labeled inactive.

Since there may be more than two non-dominated individuals attaching to one reference vector, we need to sort them within a cluster. We evaluate one individual from its proximity and distributivity. The proximity is reflected by the distance along the closest λ . The smaller the value, the closer the individual is to the true PF. The distributivity is measured by the perpendicular distance between the non-dominated individual cs and λ . This distance represents the distribution error between cs and λ . The smaller the value, the closer cs is to λ . These two criteria are integrated into the following formula.

$$\begin{aligned} P(cs) &= d_1 + \alpha d(Q(cs), \lambda), \\ &= Q(cs)\lambda^T + \alpha \times \sqrt{\|Q\|_2^2(cs) - d_1^2}, \end{aligned} \quad (10)$$

where α is a parameter that controls the proximity and distributivity. We can sort the non-dominated individuals in ascending order based on the compromise value of cs .

For the second-stage clustering, we also need guidance for the search of the dominated individuals towards proximity and distributivity. Some characteristics of the second-stage clustering are summarized below.

- (i) For each cluster in the first stage, the top individual is taken as the center of the second-stage clustering.
- (ii) The dominated individuals are assigned to the closest center according to the Euclidean distance between the individuals and the centers.

- (iii) The dominated individuals in the same cluster are sorted in ascending order based on the comparison of their closest value to the center.

4.3.2. Population Selection. We propose a cyclic selection to distribute a new population close to the true PF evenly. The idea behind this is that, by selecting the feasible individuals having the best values in different clusters, we have a higher chance of obtaining a set of solutions with good proximity and distributivity. More specifically,

- (i) For each first-stage clustering, the feasible head of the sorted non-dominated individual's list is selected.
- (ii) The feasible head of the sorted dominated individuals' list is selected in order.
- (iii) If the number of the selected individuals is less than the population size, other feasible non-dominated individuals are selected in order.

5. Experiments and Analysis

5.1. Experiment Design

5.1.1. Dataset. Given a workflow with a set of tasks, there are concrete services with similar functions but different QoS values for each task. Therefore, there are a large number of composition instances. For each test case, the concrete services are randomly assigned using the QWS dataset (<https://qwsdata.github.io/>), which records the QoS measurements of real-world web services. We focus on the response time, availability, throughput, successability, and reliability attributes. All experimental results are collected on a 3.4 GHz PC with 8 GB RAM.

5.1.2. Comparative Approaches

- (i) MODE: it is a basic multi-objective differential evolution algorithm for verifying the impact of the dual strategy and fine-grained search on the performance of our proposed algorithm.
- (ii) MOGP: A multi-objective genetic programming algorithm is proposed in [34]. It is a powerful evolutionary metaheuristic to find the best trade-offs between more than two objectives.
- (iii) MS-DABC: An improved artificial bee colony algorithm is proposed in [3]. It has a competitive performance produced by cooperating with a synergistic mechanism, a diversity maintenance strategy, and a well-maintained external achieve with the artificial bee colony algorithm.
- (iv) NSGA-III-DDR: An improved evolutionary multi-objective optimization algorithm using reference-point based non-dominated sorting approach is proposed in [35]. It provides a distance dominance relationship in NSGA-III. The algorithm not only considers the diverse solutions but also retains good convergence.

5.1.3. Parameter Setting. The shared parameters for all algorithms are as follows: the number of epochs is set to 200; the population size N is set to 100.

The individual parameters for each algorithm are as follows. For the EnMODE, the values of F_1 and F_2 are respectively set to 0.8; the values of F_3 and F_4 are respectively set to 0.4; the parameter that controls the division on the objective axis is $H=8$; the parameter that controls the proximity and distributivity is $\alpha = 5$; The maximum number of iterations is $\text{itermax}=200$. For the comparative algorithms, the crossover rate and mutation rate are set to 0.7 and 0.3, respectively.

5.1.4. Evaluation Metrics. GD [36] measures the proximity degree of the obtained solutions set toward the true Pareto-optimal set. It is computed using the quadratic mean of the Euclidean distances from N compositions in the obtained solutions set to the closest composition in the true Pareto-optimal set. Based on [37], we identify the true Pareto-optimal set by selecting N compositions from the union of the obtained solutions from the EnMODE and the comparative algorithms. The formula of GD is computed as follows.

$$GD(CS, CS^*) = \frac{\sqrt{\sum_{i=1, cs_i \in CS}^N d^2(cs_i, CS^*)}}{N}, \quad (11)$$

where CS and CS^* represent the obtained solutions set and the true Pareto-optimal set, respectively. $d(cs_i, CS^*)$ is the Euclidean distance from cs_i to the closest composition in CS^* . The smaller the value of GD, the better the proximity degree.

SP [38] measures the uniformity of the obtained solutions set on distribution. It is computed using the distance variance between compositions in the obtained solutions set.

$$SP(CS) = \sqrt{\frac{\sum_{i=1}^N (d(cs_i, cs_j) - \bar{d})^2}{N-1}}, \quad (12)$$

where $d(cs_i, cs_j)$ represents the Euclidean distance from cs_i to the closest cs_j , with $cs_j \in CS$ and $j \neq i$. $\bar{d} = (\sum_{i=1}^N (d(cs_i, cs_j))/N)$. The smaller the value of SP, the more uniform the obtained solutions set.

The size of the evaluation metrics is affected by the number of abstract services and concrete services. Thus, different parameter configurations are given as follows: the size of abstract services varies from 5 to 50 with a step of 5, and the size of concrete services varies from 100 to 1000 with a step of 100.

5.2. Analysis of Experimental Results

5.2.1. Analysis of the Proximity Problem. Figure 2 shows the GD values obtained by the five algorithms on 10 test cases with different numbers of concrete services. From the figure, we can see that the GD values obtained by the EnMODE on all test cases are smaller than the GD values obtained by other algorithms. More specifically, The EnMODE has

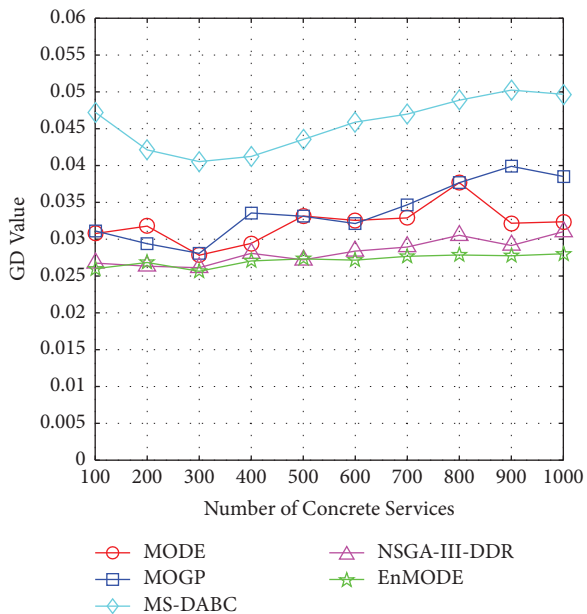


FIGURE 2: GD value on test cases with different numbers of concrete services.

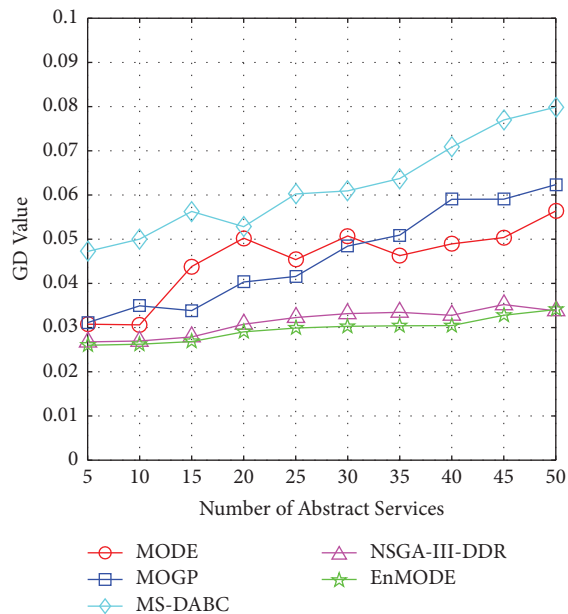


FIGURE 3: GD value on test cases with different numbers of abstract services.

smaller GD values than the baseline MODE. Meanwhile, the GD values obtained by the EnMODE on test cases of 100, 300, 400, 600, 700, 800, 900, and 1000 concrete services are slightly smaller than those obtained by NSGA-III-DDR, while the GD values obtained by the EnMODE on test cases of 200 and 500 concrete services are significantly smaller than those obtained by the NSGA-III-DDR. The GD values obtained by the EnMODE are significantly smaller than the GD values obtained by the MOGP and MS-DABC. As the number of concrete services grows, the GD values for each algorithm increase at a slow growth rate. The EnMODE has the slowest rate of increase. Figure 3 displays the GD values on 10 test cases with different numbers of abstract services. The figure shows that the EnMODE has smaller GD values than the baseline MODE, MOGP, NSGA-III-DDR, and MS-DABC in all test cases. Specifically, the EnMODE has slightly smaller GD values on first night test cases than the NSGA-III-DDR. Meanwhile, the EnMODE has significantly smaller GD values than the baseline MODE, MOGP, and MS-DABC. As the number of abstract services increases, there is a growing gap between EnMODE and other algorithms.

Through the experimental results on different test cases, we can see that the EnMODE has better proximity in terms of GD values by comparing the baseline MODE. It can be seen that the EnMODE with the dual strategy and fine-grained search can better improve the performance of the algorithm, especially in the proximity problem.

The experimental results on different test cases show that the EnMODE has the best GD values by comparing the competing approaches MOGP, NSGA-III-DDR, and MS-DABC. The reason for our analysis may be that as populations evolve, there is less evolutionary pressure. However, the dual strategy proposed in Section 4 provides sufficient evolutionary pressure for the population at different stages.

In the early stage, the exploration ability of “rand/2/bin” is developed with a high probability. More and more services would be utilized to compose different value-added services, which result in a more high-quality composition with a higher chance of success. As the number of iterations increases, “current-to-best/1/bin” is gradually utilized to search around the potential high-quality region. The optimized information guides the exploration toward the potential optimized region, which makes EnMODE possible to converge to true PF. “rand/2/bin” is used simultaneously to explore more new compositions. The usage of “current-to-best/1/bin” would be enlarged during the posterior stage so that the creation operator uses more information to generate high-quality compositions. Even if the search status changes with the increase of concrete services and abstract services, the EnMODE provides sufficient evolution pressure to reduce the influence. Because MOGP, NSGA-III-DDR, and MS-DABC ignore the evolutionary pressure, they have worse proximity than the EnMODE.

5.2.2. Analysis of Uniform Distribution Problem. The SP results of each algorithm over 10 test cases for different numbers of concrete services are shown in Figure 4. We can see that the EnMODE has optimal SP values on test cases of 300, 400, 500, 700, 800, 900, and 1000 concrete services, and the NSGA-III-DDR has optimal SP values on test cases of 100, 200, and 600 concrete services. In general, the EnMODE algorithm has the strongest competitiveness. In contrast, the baseline MODE, MOGP, NSGA-III-DDR, and MS-DABC have worse SP values. As the number of concrete services grows, the SP values for each algorithm increase, but the growth rate of the EnMODE is less than other algorithms. The SP results of each algorithm over 10 test cases for different numbers of abstract services are displayed in

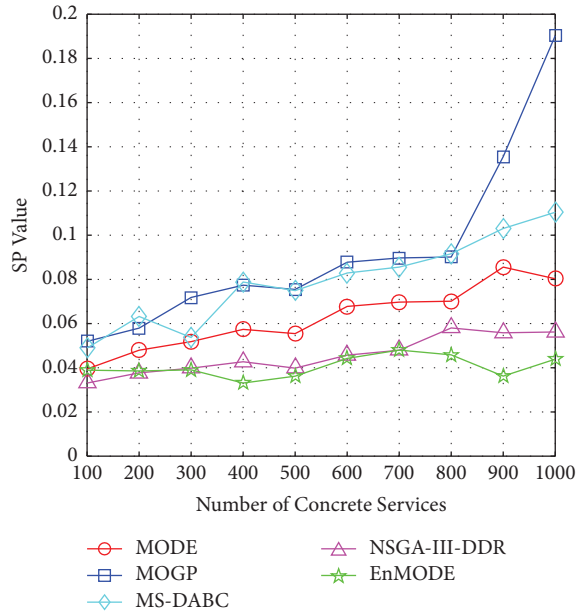


FIGURE 4: SP value on test cases with different numbers of concrete services.

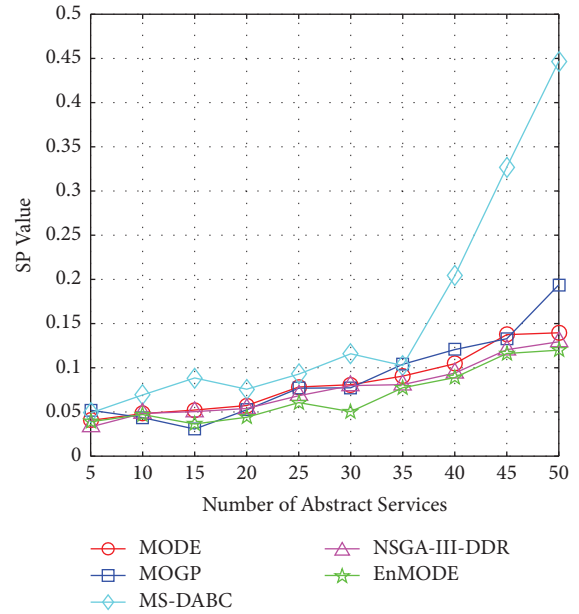


FIGURE 5: SP value on test cases with different numbers of abstract services.

Figure 5. From the figure, EnMODE outperforms its rivals. More specifically, the EnMODE has slightly smaller SP values than the NSGA-III-DDR, while the MODE, MOGP, and MS-DABC have significantly bigger SP values than the EnMODE. As the increase of abstract services, the SP values for EnMODE and NSGA-III-DDR, in general, increase at a low rate, while the values for MODE, MOGP, and MS-DABC increase quickly in all cases. Especially the growth rate of the EnMODE is less than the NSGA-III-DDR.

We can conclude from the experimental results on different test cases that the EnMODE has a more uniform distribution in terms of SP values by comparing the baseline MODE. It can be seen that the EnMODE with the dual strategy and fine-grained search can better improve the performance of the algorithm, especially in the uniform distribution problem.

From the experimental results, we also conclude that the EnMODE has the best uniform distribution in terms of SP values. We also found out that the numbers of concrete services and abstract services would influence the sizes of SP values. Still, the EnMODE has a narrower range of variation in SP values than other algorithms. As stated in Section 4, the reference vector neighbor search uses two-stage clustering to downsize the problem and then conducts a fine-grained search. The first stage of clustering divides the non-dominated compositions under the guidance of the reference vector to cause similar and distinct compositions to gather. The second stage of clustering assembles similar and distinct dominated compositions around the elites in the first stage. Two-stage clustering achieves natural-organized decomposition. A cyclic selection is used to make the distribution of the new generation close to the true PF evenly. Even if the sizes of concrete services and abstract services grow, the EnMODE provides a fine-grained search to make the distribution of the obtained solutions set over the whole

extent of the current PF more uniform. The reason why the NSGA-III-DDR algorithm has slightly poor uniformity may have a good distance dominance relationship. The MOGP and MS-DABC achieve an evolutionary process by the genetic operators, which makes them difficult to make the solutions set close to the true PF evenly.

5.3. *Summary of Results.* Based on the evaluative results of the experiments, we have verified that the EnMODE algorithm finds a smaller set of solutions with better proximity and distributivity. Compared with MODE, NSGA-III-DDR, MS-DABC, and MOGP, the reference vector neighbor search gives EnMODE better proximity and distributivity. In addition, with the increase of concrete and abstract services, the influence of EnMODE is less than other algorithms.

6. Conclusion and Future Work

This paper proposes a novel multi-objective differential evolution algorithm as the search scheme. The proposed approach implements the natural-organized decomposition of MOP, and guides the search of multiple sub-problems around the active reference vector and high-quality non-dominated compositions. Experimental results verify that the proposed approach is more likely to find a representative set of solutions with proximity and distributivity.

This work is expected to investigate the impact of the number of QoS objectives on the optimization problem of MOSC. Furthermore, the proposed approach is improved to adapt to the workflow change.

Data Availability

The experimental data used to support the findings of this study are available upon request to the author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. U22A2026), the Project Funded of the China Postdoctoral Science Foundation (No. 2021M693727), the Natural Science Foundation of Chongqing (No. 2022NSCQ-MSX1809), the Special Fund of the Chongqing Postdoctoral Science Foundation (No. XmT2020180), the Science and Technology Research Program of Chongqing Municipal Education Commission (No. KJQN202100515), and the Foundation Projection of Chongqing Normal University (No. 21XLB003).

References

- [1] T. H. Tan, M. Chen, J. Sun et al., "Optimizing selection of competing services with probabilistic hierarchical refinement," in *Proceedings of the 38th International Conference on Software Engineering, ICSE*, pp. 85–95, Austin, TX, USA, May 2016.
- [2] F. Chen, M. Li, and H. Wu, "GACRM: a dynamic multi-attribute decision making approach to large-scale web service composition," *Applied Soft Computing*, vol. 61, pp. 947–958, 2017.
- [3] J. Zhou, X. Yao, Y. Lin, F. T. S. Chan, and Y. Li, "An adaptive multi-population differential artificial bee colony algorithm for many-objective service composition in cloud manufacturing," *Information Sciences*, vol. 456, pp. 50–82, 2018.
- [4] A. S. da Silva, H. Ma, Y. Mei, and M. Zhang, "A hybrid memetic approach for fully automated multi-objective web service composition," in *Proceedings of the 2018 IEEE International Conference on Web Services (ICWS)*, pp. 26–33, Seattle, WA, USA, June 2018.
- [5] B. Tan, H. Ma, Y. Mei, and M. Zhang, "Evolutionary multi-objective optimization for web service location allocation problem," *IEEE Transactions on Services Computing*, vol. 14, no. 2, pp. 458–471, 2021.
- [6] M. Cremene, M. Suci, D. Pallez, and D. Dumitrescu, "Comparative analysis of multi-objective evolutionary algorithms for qos-aware web service composition," *Applied Soft Computing*, vol. 39, pp. 124–139, 2016.
- [7] D. Guo, X. Wang, K. Gao, Y. Jin, J. Ding, and T. Chai, "Evolutionary optimization of high-dimensional multi-objective and many-objective expensive problems assisted by a dropout neural network," *IEEE Transactions on Systems, Man, and Cybernetics - Scimago*, vol. 52, no. 4, pp. 2084–2097, 2022.
- [8] W. Li, L. He, and Y. Cao, "Many-objective evolutionary algorithm with reference point-based fuzzy correlation entropy for energy-efficient job shop scheduling with limited workers," *IEEE Transactions on Cybernetics*, vol. 52, no. 10, pp. 10721–10734, 2022.
- [9] Y. J. Cruz, M. Rivas, R. Quiza, A. Villalonga, R. E. Haber, and G. Beruvides, "Ensemble of convolutional neural networks based on an evolutionary algorithm applied to an industrial welding process," *Computers in Industry*, vol. 133, Article ID 103530, 2021.
- [10] Y. Chen, T. Shi, H. Ma, and G. Chen, "Automatically design heuristics for multi-objective location-aware service brokering in multi-cloud," in *Proceedings of the IEEE International Conference on Services Computing (SCC)*, pp. 206–214, Barcelona, Spain, July 2022.
- [11] Y. Wang, S. Wang, S. Gao, X. Guo, and B. Yang, "Adaptive multi-objective service composition reconfiguration approach considering dynamic practical constraints in cloud manufacturing," *Knowledge-Based Systems*, vol. 234, Article ID 107607, 2021.
- [12] Z. Yang, Y. Ding, Y. Jin, and K. Hao, "Immune-endocrine system inspired hierarchical coevolutionary multiobjective optimization algorithm for iot service," *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp. 164–177, 2020.
- [13] H. Chen, G. Wu, W. Pedrycz, P. N. Suganthan, L. Xing, and X. Zhu, "An adaptive resource allocation strategy for objective space partition-based multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics System*, vol. 51, no. 3, pp. 1–16, 2019.
- [14] L. Xing, R. Wu, J. Chen, and J. Li, "Handling irregular many-objective optimization problems via performing local searches on external archives," *Mathematics*, vol. 11, no. 1, p. 10, 2023.
- [15] P. Dai, K. Liu, L. Feng et al., "Temporal information services in large-scale vehicular networks through evolutionary multi-objective optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 218–231, 2019.
- [16] H. Seada and K. Deb, "A unified evolutionary optimization procedure for single, multiple, and many objectives," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 358–369, 2016.
- [17] G. Dhiman, M. Soni, H. M. Pandey, A. Slowik, and H. Kaur, "A novel hybrid hypervolume indicator and reference vector adaptation strategies based evolutionary algorithm for many-objective optimization," *Engineering with Computers*, vol. 37, no. 4, pp. 3017–3035, 2021.
- [18] Q. Lin, Y. Ma, J. Chen et al., "An adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies," *Information Sciences*, vol. 430–431, pp. 46–64, 2018.
- [19] A. Kumar, R. K. Misra, D. Singh, and S. Das, "Testing A multi-operator based differential evolution algorithm on the 100-digit challenge for single objective numerical optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2019*, pp. 34–40, Wellington, New Zealand, June 2019.
- [20] E. V. Altay and B. Alatas, "Differential evolution and sine cosine algorithm based novel hybrid multi-objective approaches for numerical association rule mining," *Information Sciences*, vol. 554, pp. 198–221, 2021.
- [21] H. Cheng, M. Zhong, and J. Wang, "Diversified keyword search based web service composition," *Journal of Systems and Software*, vol. 163, Article ID 110540, 2020.
- [22] J. Li, G. Fan, M. Zhu, and Y. Yan, "Pre-joined semantic indexing graph for qos-aware service composition," in *Proceedings of the 2019 IEEE International Conference on Web Services (ICWS)*, pp. 116–120, Milan, Italy, July 2019.
- [23] C. Jatoth, G. R. Gangadharan, and R. Buyya, "Optimal fitness aware cloud service composition using an adaptive genotypes evolution based genetic algorithm," *Future Generation Computer Systems*, vol. 94, pp. 185–198, 2019.
- [24] X. Wang, X. Xu, Q. Z. Sheng, Z. Wang, and L. Yao, "Novel artificial bee colony algorithms for qos-aware service selection," *IEEE Transactions on Services Computing*, vol. 12, no. 2, pp. 247–261, 2019.

- [25] P. Rodríguez-Mier, M. Mucientes, and M. Lama, "Hybrid optimization algorithm for large-scale qos-aware service composition," *IEEE Transactions on Services Computing*, vol. 10, no. 4, pp. 547–559, 2017.
- [26] J. I. Siebert, J. Cao, Y. Lai, P. Guo, and W. Zhu, "LASEC: a localized approach to service composition in pervasive computing environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 7, pp. 1948–1957, 2015.
- [27] 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 Transactions on Services Computing*, vol. 9, no. 5, pp. 806–817, 2016.
- [28] E. Martín, A. Cervantes, Y. Saez, and P. Isasi, "IACS-HCSP: improved ant colony optimization for large-scale home care scheduling problems," *Expert Systems with Applications*, vol. 142, Article ID 112994, 2020.
- [29] H. Wang, X. Chen, Q. Wu et al., "Integrating reinforcement learning with multi-agent techniques for adaptive service composition," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 12, no. 2, pp. 1–42, 2017.
- [30] S. Peng, H. Wang, and Q. Yu, "Estimation of distribution with restricted Boltzmann machine for adaptive service composition," in *Proceedings of the 2017 IEEE International Conference on Web Services (ICWS)*, pp. 114–121, Honolulu, Hawaii, USA, June 2017.
- [31] H. Wang, S. Peng, and Q. Yu, "A parallel refined probabilistic approach for qos-aware service composition," *Future Generation Computer Systems*, vol. 98, pp. 609–626, 2019.
- [32] A. Palade and S. Clarke, "Collaborative agent communities for resilient service composition in mobile environments," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 876–890, 2022.
- [33] Y. Xie, J. Qiao, D. Wang, and B. Yin, "A novel decomposition-based multiobjective evolutionary algorithm using improved multiple adaptive dynamic selection strategies," *Information Sciences*, vol. 556, pp. 472–494, 2021.
- [34] H. Wang, A. Ouni, M. Kessentini, B. R. Maxim, and W. I. Grosky, "Identification of web service refactoring opportunities as a multi-objective problem," in *Proceedings of the IEEE International Conference on Web Services (ICWS)*, pp. 586–593, San Francisco, CA, USA, July 2016.
- [35] Q. Gu, Q. Xu, and X. Li, "An improved NSGA-III algorithm based on distance dominance relation for many-objective optimization," *Expert Systems with Applications*, vol. 207, Article ID 117738, 2022.
- [36] S. Jiang, Y. Ong, J. Zhang, and L. Feng, "Consistencies and contradictions of performance metrics in multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2391–2404, 2014.
- [37] Y. Xiang, Y. Zhou, X. Yang, and H. Huang, "A many-objective evolutionary algorithm with pareto-adaptive reference points," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 99–113, 2020.
- [38] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: a survey," *ACM Computing Surveys*, vol. 52, no. 2, pp. 1–38, 2019.