Hindawi
Mobile Information Systems
Volume 2022, Article ID 5444055, 13 pages
https://doi.org/10.1155/2022/5444055

Research Article

A Decomposition and Dominance-Based Multiobjective Artificial Bee Colony Algorithm for Multiple Sequence Alignment

Lei Ye

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China

Correspondence should be addressed to Lei Ye; hunnan214@163.com

Received 10 January 2022; Accepted 23 February 2022; Published 24 March 2022

Academic Editor: Jianhui Lv

The multiple sequence alignment (MSA) problem is essential in biological research for finding a specific relationship between the biologic sequences and their function. This study proposes a decomposition and dominance-based multiobjective artificial bee colony optimization algorithm for MSA (MOABC/D-MSA). MOABC/D-MSA uses three kinds of searching strategies to obtain a group of nondominated solutions with high quality and diversity of an MSA problem. A decomposition-based employed bee strategy is proposed to search for high-performance solutions of the MSA, while insuring their diversity. A nondominated sorting-based onlooker strategy searches for the solutions near the Pareto front (PF) to guide the subsequent searching. The scout bee strategy facilitates the algorithm to get out of the local optimal. A comparative experiment is implemented on BAliBASE 3.0, a benchmark for MSA algorithms. Experimental results show that the proposed algorithm has competitive performance with state-of-the-art metaheuristic algorithms. Furthermore, nondominated solutions of MOABC/D-MSA have a more uniform distribution in the objective space.

1. Introduction

The multiple sequence alignment (MSA) problem aims to find shared fragments among three or more sequences. In general, the sequences to be aligned are biological sequences, such as nucleotides, amino acids, and proteins. The results of the alignment are a set of sequences that are transformed from the original sequences by inserting several gaps. The sequences in the alignment solution have the same length. From the alignment results, researchers could identify regions that exist the same elements among the sequences. These regions could reveal potential relations among the input sequences, such as the evolutionary relationship among the sequences [1] and the influence of special sites on structure and function.

The MSA problem has been proven to be an NP-complete problem with the sum-of-pairs (SP) metric [2] and NP-hard for most of the existing metrics [3]. Therefore, researchers have paid a lot of attention to developing an effective and efficient approach for MSA. Different kinds of algorithms are proposed to solve the MSA problem.

Literature [4] finds six main groups of different approaches for dealing with biological MSA problems: (1) exact methods, (2) progressive methods, (3) consistency-based methods, (4) iterative methods, (5) evolutionary algorithms, and (6) structure-based methods.

The metaheuristic algorithms, including evolutionary algorithms and swarm intelligence algorithms, have shown competitive performance in optimizing MSA problems. Literature [4] proposed a hybrid multiobjective metaheuristic for MSA, which combines the shuffled frog-leaping optimization algorithm with the fast and accurate Kalgin algorithm. Literature [5] proposes a characteristic-based framework for MSA, which extracts the characteristics of unaligned input sequences and aligns the sequences with the specific configuration according to the characteristics. Ten kinds of characteristics are considered in this study; they are divided into three groups. A multiobjective evolutionary algorithm for MSA (MOMSA) is proposed in reference [2]. MOMSA is a biobjective aligner based on the framework of MOEA/D. The Tchebycheff approach is adopted for subproblem design. Furthermore, this study proposes a tree-

based initialization method and a gap-reinserting-based mutation operator. Literature [6] proposes a quantum-inspired heuristic optimization method for MSA, where a quantum-inspired GVN routine is designed. Literature [1] proposes a multiobjective formulation for MSA for inferring the evolutionary history relating the sequences known as phylogenetic trees. ProbPFP [7] combines the partition function and the hidden Markov model (HMM). The parameters of the HMM are optimized by particle swarm optimization (PSO). Literature [8] proposes an algorithm that employs sparse approximation to reduce the computational cost for the relaxation. Literature [9] proposes a hybrid artificial bee colony (ABC) optimization algorithm for MSA. This algorithm performs a single-point crossover for the employed bee phase. It performs a multiple mutation operator that contains four kinds of mutations operators for the onlooker bee phase. The Kalign2 algorithm is implemented in the scout bee phase to align a segment of the sequences.

Early literature evaluates the alignments by one score function. However, when optimizing the MSA problem, researchers have more than one requirement for aligned sequences. For example, researchers hope to find as many similar fragments as possible. Meanwhile, they want to insert gaps as much as possible. To meet the different requirements of researchers, the MSA problem is designed to contain multiple optimization objectives. Researchers have proposed different definitions of multiobjective MSA problems. Reference [4] adopts the weighted sum-of-pairs function with affine gap penalties (WSP) and the number of conserved total column (TC) score. Reference [5] uses the Q-score (i.e., sum-of-pairs (SP) score) and total column (TC) score as the objective functions. In reference [2], the MSA problem is a biobjective minimization problem, where the first objective function is the scoring function that minimizes the number of gaps and the second objective function is the opposite of the SP function. Four new objectives are proposed in reference [1]as follows: nongap columns for the calculation of entropy, the similarity of columns containing one or more gaps, the similarity of columns containing no gap, and the number of consecutive gaps. Reference [8] develops a relaxed formulation for the MSA problem based on a regression-coding framework.

Since the optimization objectives in a multiobjective MSA are conflicted in most cases, it is hard to optimize all objectives simultaneously. In practice, the optimization algorithm solves the multiobjective problem by working out a set of solutions that reflects the trade-off of the objectives. The relationship between the solutions of the set of solutions is Pareto nondominated.

Metaheuristics have been proven to be effective in optimizing multiobjective problems [10–13]. In this study, a typical metaheuristic algorithm, ABC [14], is adopted to optimize a multiobjective MSA problem. ABC is one of the popular metaheuristic algorithms. It mimics the behaviors of three kinds of bees: employed bees, onlooker bees, and scout bees. The earliest ABC for MSA known in this research is a single-objective algorithm [15]. The motivation for using ABC in this research is that its searching strategy is suitable for solving a complicated multiobjective problem. The ABC uses different kinds of searching strategies; therefore, it can balance the convergence speed and solution quality. Furthermore, the coupling between the three stages of the ABC is low, and the algorithm designer can design targeted strategies for solving MSA problems according to their needs. When optimizing a multiobjective MSA problem, the algorithm needs to handle two tasks: making the solutions converge to the Pareto frontier (PF) of the problem and ensuring the uniform distribution of solutions. The decomposition strategy is adopted in this study. A decomposition-based algorithm could obtain a set of evenly distributed nondominated solutions. It has shown strong performance in approximating the shape of the PF of the multiobjective problem. Furthermore, to improve the effectiveness and efficiency of the algorithm, the ABC needs to be well designed. For the proposed ABC, the employed bees are utilized for making the solutions converge to the PF and be distributed uniformly; the task of the onlooker bees is accelerating the convergence speed of the algorithm; the scout bees aim to prevent the algorithm from falling into local optimums, which is a common phenomenon during the iterations of ABC.

This study proposes a novel ABC algorithm for multiobjective MSA based on decomposition and dominance (MOABC/D-MSA). MOABC/D-MSA uses the decomposition-based multiobjective optimization strategy to ensure the diversity of solutions. Therefore, it can provide MSA users the information about the shape of the PF, which is essential to decision-making. The main contribution of this study is as follows:

(1) A novel ABC algorithm for MSA is proposed. The proposed algorithm considers both the solution quality and diversity. The employed bee stage achieves even distribution of solutions while optimizing the MSA problem. The onlooker bee stage can obtain high-quality solutions based on superior solutions found by employed bees. The scout bee could facilitate the algorithm in avoiding local optimums.

(2) A decomposition-based employed bee searching strategy is employed for optimizing the MSA problem. The proposed algorithm decomposes the multiobjective MSA problem into several scalarized subproblems. The solutions of subproblems are used to construct a nondominated solution set. This strategy allows the algorithm to obtain a group of solutions distributed in the objective space uniformly.

(3) A nondominance sorting-based onlooker bee searching strategy is proposed. This strategy allows the proposed algorithm to improve the quality of alignments by utilizing high-quality solutions that have been searched.

(4) An experimental study on an MSA benchmark is implemented. The experiment compares the proposed algorithm with state-of-the-art evolutionary and metaheuristic algorithms.

The remaining parts of this study are organized as follows: the second section introduces the definition of the multiobjective MSA problem in the proposed study. Section 3 describes the design and implementation of the proposed MOABC/D-MSA. The fourth section compares the MOABC/D-MSA with state-of-the-art metaheuristics on BAliBASE 3.0, a benchmark MSA test suite. The last section summarizes the proposed work and predicts the research direction of metaheuristic algorithms for MSA.

2. Problem Definition

In this study, the multiobjective MSA problem is defined as a three-objective optimization problem.

There are three objectives in the problem: single structure induced evaluation (STRIKE), percentage of totally conserved columns (%TC), and percentage of non-gaps (% nonGap). STRIKE aims to maximize the accuracy of the alignment. Maximizing %TC ensures there are more columns that the residues are exactly the same, that is, more conserved or special regions within an alignment. Maximizing the %nonGap encourages the aligner to reduce the number of gaps in the aligned sequences. The MSA problem is represented by the mathematical form as shown in the following equation:

$$\text{maximize } F(S) = (\text{STRIKE}(S), \%TC(S), \%\text{nonGap}(S)). \tag{1}$$

Strike evaluates the accuracy of an alignment based on structural information of, at least, one sequence of the alignment. This structural information is commonly retrieved from the Protein Data Bank [16].

Using the structural information as a source for amino acid frequencies and contacts, a log-odds contact matrix is estimated by measuring the ratio between the frequency of each possible contact and its expectation, given the background frequency of each single amino acid. Given any pair of amino acids i and j, the score for their contacts can be estimated as follows:

$$M_{ij} = 10 \times \ln\left(\frac{f_{ij}}{f_i f_j}\right), \tag{2}$$

where f_{ij} is the frequency of contacts involving i and j across all observed *residue-residue* contacts and f_i and f_j are the single residue frequencies in the dataset considered.

%TC takes into account the number of columns that are fully aligned with exactly the same compound. TC is defined as shown in the following equation:

$$\%TC(S) = 100 \sum_{l=1}^{L} \frac{\text{totalColumn}(S_l)}{L}, \tag{3}$$

where S_l is the l^{th} column of S, $S_l = s_{il} \forall i = 1, \ldots, k$, and $\text{totalColumn}(S_l)$ is defined as follows:

$$\text{totalColumn}(S_l) = \begin{cases} 1, & \text{if } s_{il} = s_{1l} \forall i = 2, \ldots, k \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

%nonGap measures the number of residues with respect to the number of gaps into the alignment. This objective function is shown in the following equation:

$$\%\text{nonGap}(S) = 100 \sum_{i=1}^{k} \sum_{j=1}^{L} \frac{\text{is NotGap}(s_{ij})}{k \times L}, \tag{5}$$

where s_{ij} represents the symbol in the j^{th} position of the i^{th} sequence in the alignment S. The function *isNotGap* for a specific residue is defined in the following equation:

$$\text{isNotGap}(\text{residue}) = \begin{cases} 1, & \text{if residue} = '' - '' \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

where the symbol "—" is a placeholder that is used to align subsequences.

3. MOABC/D-MSA

3.1. Representation of Individuals. Since the solution for an MSA problem is a sequence of characters and gaps, this study adopted an encoding strategy that records the positions of gaps for each sequence. The adopted representation uses the format (begin, end) to store the position of a gap or several consecutive gaps.

Figure 1 illustrates the solution representation in this study. The solution s has four regions that exist gap or gaps: the fifth character, the eleventh to the fourteenth characters, the eighteenth character, and the twentieth to twenty-first characters. This solution is represented by the s' in Figure 1.

3.2. Crossover and Mutation Operators. The search behaviors of the bees are based on crossover and mutation operators. Similar to most of the metaheuristics for MSA, this study employs a single-point crossover operator. For two solutions p_1 and p_2 that $\text{len}(p_1) = \text{len}(p_2) = l_p$, the crossover operator generates a random integer d between zero and l_p. Then each sequence of p_1 is divided into two parts after the d^{th} position, assuming that they are k sequences, p_1 is divided to $\{p^1_{1,1}, p^2_{1,1}, \ldots, p^1_{1,k}, p^2_{1,k}\}$. For p_2, each sequence is cut after the last character of the corresponded sequence of p_1 that is not gap.

After the division, a child solution c_1 is constructed by connecting the first part of p_1 and the last part of p_2. The other child solution c_2 is constructed by connecting the last part of p_2 and the last part of p_1. Since all sequences should have the same length, the crossover operator should adjust the new sequences by inserting gaps into them. The length of all sequences of the child solutions is the length of the longest sequence. This length is calculated according to the following equation:

$$l_{\text{new}} = \max\left(\max\left(p^1_{1,i} + p^2_{2,i}\right)\right), \quad i = 1, \ldots, k. \tag{7}$$

For a sequence whose length is shorter than l_{new}, the crossover operator inserts gaps between the two segments from p_1 and p_2 to make the sequence's length reach l_{new}.

There are three kinds of mutation operators adopted in the proposed algorithm:

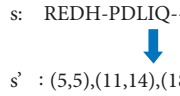

Figure 1: Illustration of individual representation.

(i) *Shift-closed gaps.* Closed gaps (i.e., a series of sequential gaps) in a sequence are randomly chosen and shifted to another position randomly. At last, gaps columns (i.e., columns that only contain gaps) are removed.

(ii) *Nongap group splitting.* A non-gap group (i.e., a series of characters between two gaps) is selected randomly. Then, this operator split the nongap group into two groups by inserting a gap at a random position. Finally, gaps columns are deleted.

(iii) *Adjacent gap groups merging.* For this operator, two adjacent gaps groups are selected and merged by shifting the nongap group between the two gap groups to the right.

3.3. Algorithm Overview. The proposed algorithm searches for the optimal alignment by simulating the behavior of bees collecting nectar. There are two kinds of bees in the proposed algorithm that performs different kinds of searching behavior: employed bees and onlooker bees. Algorithm 1 shows the framework of the proposed MOABC-MSA. First, the algorithm initializes N random food sources that represent N candidate alignments. Meanwhile, an archive NA for storing nondominated solutions is initialized as NULL. Then the algorithm executes searching procedures of employed bees, onlooker bees, and scout bees. The MOABC-MSA repeats the searching behaviors of two kinds of bees until it meets the stop criterion. Finally, the algorithm outputs nondominated solutions and their corresponding scores. Figure 2 shows the alignment process of the proposed algorithm.

3.4. Food Source Initialization. The initial food sources are created by MUSCLE [17], a nonmetaheuristic method. MUSCLE creates a group of precomputed alignments. This approach can reduce the execution time of the proposed algorithm.

3.5. Decomposition-Based Employed Bee Phase. The employed bees in this study perform a decomposition-based multiobjective optimization, which can obtain solutions with both high quality and diversity. The diversity is essential for the algorithm since it has an important influence on the subsequent optimization. Algorithm 2 shows the execution process of the employed bee stage.

Each employed bee is assigned a food source. The single-point crossover operation is performed. The proposed algorithm calculates the Euclidean distance between the food sources. For each food source, its nearest T food sources are defined as its neighbors. When executing the crossover operator, for the i^{th} bee, the first individual of the crossover operation is the i^{th} food source and the second individual is randomly selected from neighbors of the i^{th} food source.

```
Input: sequences to be aligned;
Output: aligned sequences;
  initialize N random food sources;
  initialize non-dominated set NA;
  evaluate food sources;
  while termination criterion is false do
    employed bees execute search behaviour;
    onlooker bees execute search behaviour;
    for each sub-problem do
      if is not updated for k iterations then
        scout bee execute search behaviour;
      end if
    end for
    update non-dominated set NA;
  end while
  output non-dominated solutions in NA;
```

Algorithm 1: Algorithm framework of MOABC/D-MSA.

After the crossover, two new candidate alignments are generated. Then the proposed algorithm performs shift-closed gaps mutation operator on the two candidate alignments and generates two offspring alignments.

Next, all offspring alignments are evaluated by the objective functions. Then the algorithm should select optimal solutions from original alignments and offspring to update the food sources.

The selection stage adopts the decomposition-based idea. The proposed algorithm generates N weight vectors ω_1 to $\omega_{\mathbf{N}}$ that are uniformly distributed in the objective space. The weight vectors are utilized in building scalarized subproblems. In this study, the penalty-based boundary intersection (PBI) approach [18] is adopted in designing subproblems. The number of subproblems equals the number of weight vectors. For the i^{th} weight vector, its corresponding subproblem is constructed as follows:

$$\text{minimize } g^{pbi}(x \mid \omega_i, z^*) = d_1 + \theta d_2, \text{ subject to } x \in \Omega, \quad (8)$$

where

$$d_1 = \frac{\left\| (\mathbf{z}^* - F(\mathbf{x}))^T \boldsymbol{\omega}_i \right\|}{\|\boldsymbol{\omega}\|} \text{ and } d_2 = \left\| F(\mathbf{x}) - (\mathbf{z}^* - d_1 \boldsymbol{\omega}_i) \right\|, \quad (9)$$

where $\mathbf{z}^*$ is the ideal point that satisfies $\mathbf{z}_i^* = \min_{i=1}^{N} f_i(\cdot)$, and each dimension of $\mathbf{z}^*$ is the minimum value among all candidate solutions.

Since each subproblem is a scalarized function, the algorithm can obtain the unique optimal solution of each subproblem in each iteration. The optimal solutions for the sub-problems are the new food sources.

3.6. Nondominated Sorting-Based Onlooker Phase. The searching behavior of onlooker bees is guided by the searching results of employed bees.

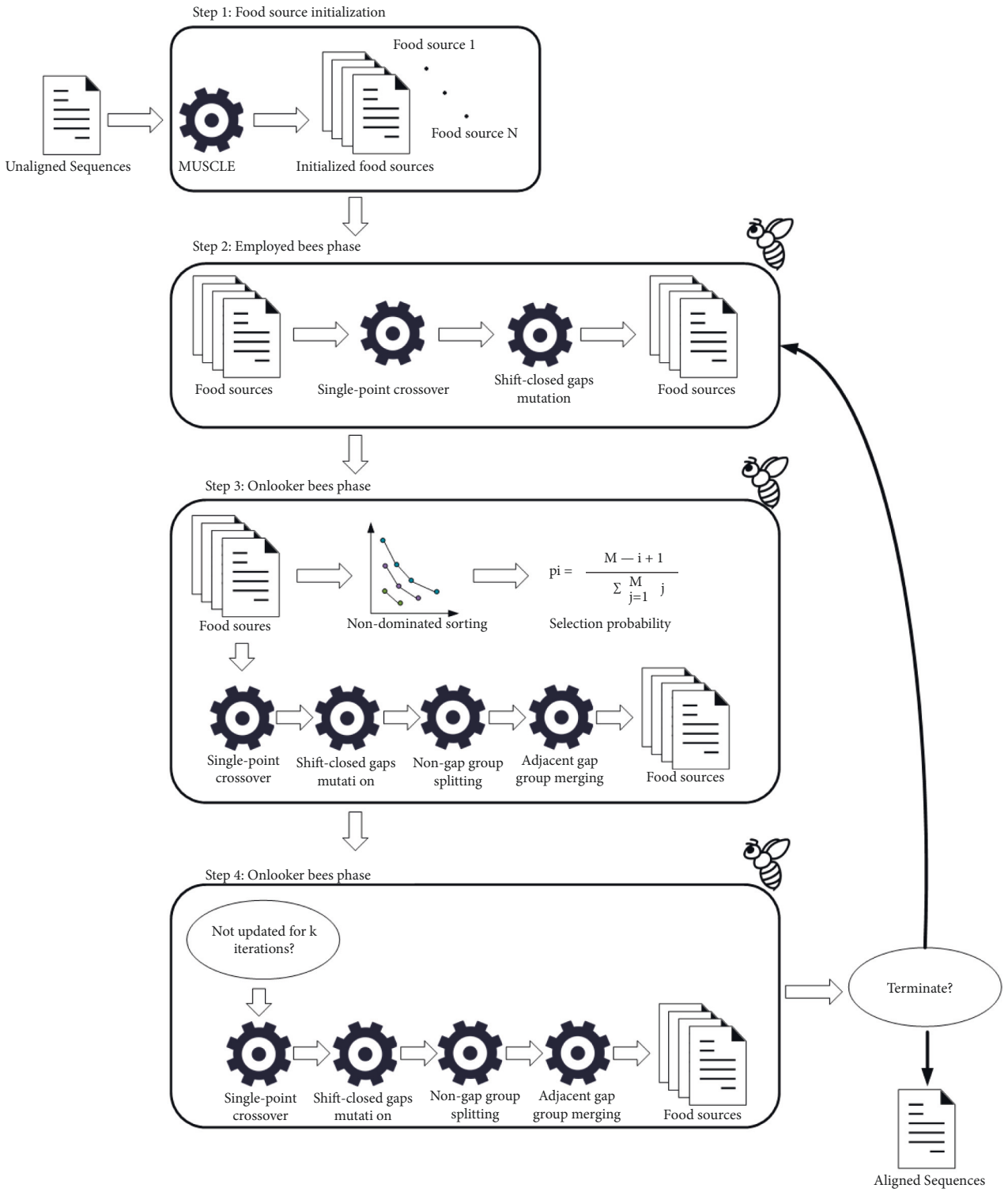

Figure 2: Alignment process of the proposed algorithm.

To guide the food sources converge to the PF, the onlooker bees prefer to exploit high-quality solutions founded by employed bees. Meanwhile, the proposed algorithm should maintain the diversity of food sources. The diversity of food sources is important for fitting the PF. Furthermore, maintaining the diversity of food sources could help the proposed algorithm to avoid being trapped in local optimal.

At this stage, each onlooker selects a food source to search. Since the employed bees have obtained the quality information of food sources, each onlooker bee selects a food

```
Input: N food sources;
Output: N new food sources;
  initialize ω1 to ωN;
  calculate neighborhood of each food source;
  while termination criterion is false do
    for each employed bee do
      perform single-point crossover operation;
      perform shift-closed gaps mutation operation;
    end for
    evaluate offspring solutions;
    merge food sources and offspring solutions;
    find optimal solution of each sub-problem;
    update food sources;
  end while
```

Algorithm 2: Searching behavior of employed bees.

source to search based on the quality information. MOABC/D-MSA uses a nondominated ranking-based roulette wheel selection. This method ranks the food sources according to their Pareto dominance relationship. Then, it uses the roulette wheel selection to guarantee that the nondominated food sources are more likely to be selected.

Inspired by the fast nondominated sorting method [19], the proposed algorithm divides the food sources into several ranks. The first rank is nondominated solutions among all food sources. The second rank is nondominated solutions among food sources except the first rank. The rest of the food sources are sorted in the same manner.

When selecting food sources, an onlooker bee first selects a rank. Assuming that there are M ranks among the food sources, the selected probability of the i^{th} rank is calculated according to the following equation:

$$p_i = \frac{M - i + 1}{\sum_{j=1}^{M} j}. \tag{10}$$

Each onlooker bee generates a random number between zero and one. Then, it selects a specific rank according to the roulette wheel method. Finally, the onlooker bee selects a food source randomly from this rank.

After selecting a food source, each onlooker bee performs a single-point crossover operation. Then it performs the shift-closed gaps mutation, nongap group splitting mutation, and adjacent gap groups merging mutation in sequence.

Finally, similar to the employed bee phase, the new solutions are merged with the food sources, and the proposed algorithm updates the food sources by the nondominated and crowded-based sorting selection [20]. The onlooker bee stage is described in Algorithm 3.

3.7. Scout Bee Phase. The scout bees aim to facilitate the algorithm avoiding local optimum. They watch the food sources in every iteration, but they work when the optimal solution of a subproblem is not updated for more than k iterations. The scout bees perform the same crossover operator and mutation operator with the onlooker bees. For the crossover operation, the scout bees select a solution from the neighbor of the food source as the mating partner. Finally, the scout bee updates the food sources of the nonupdated subproblems. Algorithm 4 shows the searching behavior of the scout bee.

```
Input: N food sources;
Output: N new food sources;
  perform fast non-dominated sorting;
  calculate selected probability for each food source;
  select a food source for each onlooker by roulette wheel
selection;
  while termination criterion is false do
    for each employed bee do
      perform single-point crossover operation;
      perform shift-closed gaps mutation operation;
      perform non-gap group splitting;
      perform adjacent gap groups merging;
    end for
    evaluate new solutions;
    merge food sources and new solutions;
    perform non-dominated and crowded-based sorting
selection;
    update food sources;
  end while
```

Algorithm 3: Searching behavior of onlooker bees.

3.8. Complexity Analysis. According to Algorithms 2–4, the time complexity of the employed bee phase, onlooker bee phase, and scout phase are all $O(n^2)$. Therefore, the time complexity of MOABC/D-MSA is $O(n^2)$. Furthermore, the scout bee phase is not always performed during the optimization process. Meanwhile, MOABC/D-MSA does not require an external archive, and it only maintains a food source set. Therefore, its space complexity is $O(n)$.

4. Experimental Study

This section proposed a comparison study. The performance of the proposed MOABC/D-MSA is compared with metaheuristics on benchmark dataset.

4.1. Benchmark Dataset. Benchmark alignment database (BAliBASE) [21] is a benchmark dataset for evaluating the performance of algorithms for MSA problems. BAliBASE is developed by manually aligning based on 3-D structures of proteins. This study uses the 3.0 version of BAliBASE to test the proposed algorithm. BAliBASE includes thousands of challenging sequences. It is widely accepted in the research of MSA problem. Therefore, using BAliBASE as the benchmark can help researchers to investigate the performance of the proposed algorithm. There are 218 sets of sequences in BAliBASE 3.0, and they are divided into six families as follows: RV11, RV12, RV20, RV30, RV40, and RV50.

This experiment selects twenty-seven test cases from BAliBASE 3.0 to implement the comparative study between

```
Input: sequences to be aligned;
Output: aligned sequences;
  initialize N random food sources;
  initialize non-dominated set NA;
  evaluate food sources;
  while termination criterion is false do
    for each sub-problem do
      if the optimal solution is not updated for more than k iterations then
        perform single-point crossover operation;
        perform shift-closed gaps mutation operation;
        perform non-gap group splitting;
        perform adjacent gap groups merging;
      end if
    end for
    evaluate new solutions;
    merge food sources and new solutions;
    find optimal solution of each sub-problem;
    update food sources;
  end while
```

Algorithm 4: Searching behavior of scout bee.

the proposed MOABC/D-MSA and other genetic and metaheuristic algorithms. The selected instances are as follows: BB11001, BB11005, BB11018, BB11020 in RV11, BB12001, BB12013, BB12022, BB12035, BB12044 in RV12, BB20001, BB20010, BB20022, BB20033, BB20041 in RV20, BB30001, BB30008, BB30015, BB30022 in RV30, BB40001, BB40013, BB40025, BB40038, BB40048 in RV40, BB50001, BB50005, BB50010, and BB50016 in RV50.

4.2. Peer Competitors. This study compares the proposed MOABC-MSA with several state-of-the-art evolutionary or metaheuristic algorithms. The competitor algorithms includes NSGA-II [22], MOEA/D [2], GAPAM [23], MO-SAStrE [24], and HMOABC [25]. The parameters of the peer algorithms are set according to their original literature.

4.3. Static Results. This section compares the statistical results of the algorithms on STRIKE, %TC, and %nonGap. Meanwhile, the executing times of the algorithms are recorded and compared. Each algorithm runs each test cases for 30 times to avoid randomness. The termination criterion is set as 25,000 times of evaluation to guarantee the fairness of the experiment. For MOABC/D-MSA and HMOABC, both the number of employed bees and the number of onlooker bees are set to 20. For the other algorithms, the size of their populations is set to 20.

Tables 1–6 list the best value of STRIKE, %TC, and % nonGap among solutions found by the tested algorithms for each test case. Tables 1–3 list that for instances in RV11, RV12, and RV20, MOABC/D-MSA obtains the optimal STRIKE, %TC, and %nonGap on all test cases. For BB30008, both MOABC/D-MSA and MOEA/D obtain the optimal % TC. For BB30015 and BB50010, MOEA/D obtains the optimal %TC. For BB40048, results of NSGA-II obtain the best %nonGap. For the other test cases in RV30, RV40, and RV50, MOABC/D-MSA outperforms the compared algorithms in all three objectives.

Figure 3 shows the box plots of running time of the tested algorithms. Subfigures (a)–(f) exhibit the running times of the algorithms on BB11001, BB12001, BB20001, BB30001, BB40001, and BB50001, respectively. The box plot is an effective tool in showing the distribution of data. In this experiment, each box in a box plot represents the distribution of running times for the 30 runs of independently repeated experiments of a specific algorithm. There are five horizontal lines in a box, from top to bottom, the lines represent the maximum value, the three-quarter median, the median, the quarter median, and the minimum value. The circles around the box represent an unusual value. The box plots show that the running time of MOABC/D-MSA is stable, in each subfigure, the box of the MOABC/D-MSA is thin. Furthermore, the location of the MOABC/D-MSA's boxes is low, which means that the proposed algorithm consumes less time. Although MOABC/D-MSA takes a little longer time than NSGA-II and MOEA/D, the running time of the proposed algorithm is significantly better than HMOABC, GAPAM, and MO-SAStrE. The boxplots show that the MOABC/D-MSA is an effective algorithm. The efficiency is an essential performance indicator in evaluating the algorithm for MSA. An efficient algorithm can ensure that the algorithm can process a large number of biological information sequences in a short time, which is practical and valuable in practice.

4.4. Hypothesis Results. This study uses the Wilcoxon signed-rank hypothesis test [26] to investigate the difference between the performance of the MOABC/D-MSA and results of the competitors. The objective values of each solution are normalized to a real number between zero and one, and then the normalized solutions are evaluated by the IGD indicator [27]. The IGD indicator works out a scalarized

Table 1: Comparison on RV11.

Test case		MOABC/D-MSA	HMOABC	NSGA-II	MOEA/D	GAPAM	MO-SAStrE
BB11001	STRIKE	**3.25**	3.07	2.94	2.98	2.79	2.88
	%TC	**7.89**	7.48	7.40	7.37	6.84	7.42
	%nonGap	**94.84**	89.46	93.75	92.98	90.57	91.44
BB11005	STRIKE	**3.14**	3.09	2.89	2.94	2.67	2.88
	%TC	**8.04**	6.90	6.59	6.54	6.38	6.52
	%nonGap	**93.68**	87.56	92.64	90.53	88.24	83.20
BB11018	STRIKE	**3.58**	3.07	2.75	2.84	2.39	2.58
	%TC	**8.44**	7.35	7.03	6.89	5.35	5.24
	%nonGap	**94.37**	87.36	92.58	92.46	90.45	91.85
BB11020	STRIKE	**3.38**	3.25	2.93	2.86	2.37	2.25
	%TC	**7.33**	7.28	7.29	7.25	7.32	7.30
	%nonGap	**93.53**	90.45	92.37	92.59	91.43	92.50

Bold numbers indicate the optimum values.

Table 2: Comparison on RV12.

Test case		MOABC/D-MSA	HMOABC	NSGA-II	MOEA/D	GAPAM	MO-SAStrE
BB12001	STRIKE	**2.74**	2.52	2.60	2.58	2.47	2.35
	%TC	**3.79**	3.62	3.74	3.75	3.68	3.71
	%nonGap	**84.40**	80.43	82.13	81.37	78.48	79.63
BB12013	STRIKE	**2.97**	2.73	2.63	2.68	2.74	2.70
	%TC	**3.79**	2.59	3.62	3.55	2.98	2.82
	%nonGap	**84.03**	80.38	83.44	82.56	81.47	78.38
BB12022	STRIKE	**2.81**	2.63	2.72	2.69	2.54	2.60
	%TC	**3.66**	3.47	3.58	3.21	3.46	3.29
	%nonGap	**82.94**	80.65	82.56	81.53	82.08	81.32
BB12035	STRIKE	**2.76**	2.51	2.69	2.65	2.55	2.49
	%TC	**3.73**	3.50	3.71	3.68	3.41	3.59
	%nonGap	**82.30**	79.73	82.21	81.99	80.35	81.02
BB12044	STRIKE	**2.71**	2.46	2.58	2.59	2.32	2.44
	%TC	**3.74**	3.56	3.67	3.66	3.69	3.52
	%nonGap	**82.39**	79.93	81.95	80.61	78.16	77.08

Bold numbers indicate the optimum values.

Table 3: Comparison on RV20.

Test case		MOABC/D-MSA	HMOABC	NSGA-II	MOEA/D	GAPAM	MO-SAStrE
BB20001	STRIKE	**0.69**	0.67	0.54	0.58	0.42	0.51
	%TC	**0.21**	0.09	0.13	0.10	0.14	0.08
	%nonGap	**41.22**	38.85	40.06	39.57	38.30	36.94
BB20010	STRIKE	**0.48**	0.42	0.47	0.41	0.39	0.44
	%TC	**0.27**	0.22	0.25	0.23	0.18	0.20
	%nonGap	**40.18**	36.94	39.82	37.81	38.89	39.06
BB20022	STRIKE	**0.28**	0.24	0.26	0.25	0.25	0.24
	%TC	**0.21**	0.18	0.19	0.17	0.17	0.15
	%nonGap	**39.42**	38.22	38.57	38.26	37.26	37.01
BB20033	STRIKE	**0.56**	0.50	0.55	0.52	0.49	0.54
	%TC	**0.26**	0.19	0.24	0.22	0.17	0.21
	%nonGap	**41.27**	38.48	40.53	36.52	34.83	35.66
BB20041	STRIKE	**0.37**	0.32	0.34	0.35	0.29	0.31
	%TC	**0.20**	0.14	0.18	0.16	0.11	0.13
	%nonGap	**40.28**	36.53	39.39	34.55	32.17	33.68

Bold numbers indicate the optimum values.

Table 4: Comparison on RV30.

Test case		MOABC/D-MSA	HMOABC	NSGA-II	MOEA/D	GAPAM	MO-SAStrE
BB30001	STRIKE	**1.75**	1.63	1.65	1.67	1.54	1.56
	%TC	**0.33**	0.29	0.32	0.30	0.25	0.22
	%nonGap	**50.77**	43.70	49.42	50.04	44.66	42.97
BB30008	STRIKE	**1.85**	1.68	1.74	1.77	1.63	1.66
	%TC	**0.33**	0.27	0.31	**0.33**	0.25	0.22
	%nonGap	**51.40**	42.34	50.06	50.56	49.28	44.30
BB30015	STRIKE	**2.44**	2.18	2.35	2.39	2.22	2.15
	%TC	0.35	0.34	0.35	**0.36**	0.29	0.28
	%nonGap	**49.07**	43.92	48.91	47.40	43.21	40.06
BB30022	STRIKE	**2.06**	1.74	1.93	1.95	1.86	1.88
	%TC	**0.41**	0.34	0.39	0.40	0.36	0.36
	%nonGap	**48.52**	46.03	47.60	47.88	42.52	43.94

Bold numbers indicate the optimum values.

Table 5: Comparison on RV40.

Test case		MOABC/D-MSA	HMOABC	NSGA-II	MOEA/D	GAPAM	MO-SAStrE
BB40001	STRIKE	**3.50**	2.99	3.45	3.47	3.09	3.14
	%TC	**0.42**	0.23	0.36	0.33	0.27	0.29
	%nonGap	**31.13**	27.62	30.75	30.90	28.84	27.56
BB40013	STRIKE	**3.79**	3.22	3.67	3.54	3.48	3.26
	%TC	**0.33**	0.21	0.29	0.27	0.20	0.26
	%nonGap	**31.01**	27.45	29.84	29.98	27.59	25.86
BB40025	STRIKE	**3.59**	3.44	3.53	3.58	3.32	3.17
	%TC	**0.38**	0.26	0.34	0.25	0.27	0.30
	%nonGap	**29.03**	25.34	28.77	27.40	24.59	22.05
BB40038	STRIKE	**3.33**	2.98	3.21	3.25	3.08	3.04
	%TC	**0.36**	0.28	0.35	0.35	0.33	0.29
	%nonGap	**29.88**	26.36	29.24	29.34	28.37	27.75
BB40048	STRIKE	**3.47**	3.05	3.36	3.42	3.06	3.11
	%TC	**0.27**	0.22	0.24	0.25	0.18	0.20
	%nonGap	29.97	26.58	**30.13**	28.44	27.93	26.55

Bold numbers indicate the optimum values.

Table 6: Comparison on RV50.

Test case		MOABC/D-MSA	HMOABC	NSGA-II	MOEA/D	GAPAM	MO-SAStrE
BB50001	STRIKE	**2.11**	1.88	1.97	2.04	1.83	1.65
	%TC	**0.39**	0.27	0.35	0.33	0.28	0.31
	%nonGap	**70.05**	55.93	69.90	67.34	62.98	63.51
BB50005	STRIKE	**2.03**	1.48	1.95	1.88	1.57	1.62
	%TC	**0.35**	0.30	0.34	0.33	0.24	0.25
	%nonGap	**69.01**	62.48	68.45	63.44	61.50	60.24
BB50010	STRIKE	**1.89**	1.56	1.84	1.80	1.61	1.69
	%TC	0.36	0.26	0.36	**0.37**	0.25	0.32
	%nonGap	**63.33**	59.02	62.75	60.45	59.93	57.28
BB50016	STRIKE	**1.89**	1.37	1.88	1.67	1.34	1.42
	%TC	**0.32**	0.19	0.30	0.31	0.24	0.25
	%nonGap	**64.01**	58.99	63.90	61.52	59.31	60.04

Bold numbers indicate the optimum values.

score for each nondominated solution set. Finally, the IGD value of each solution set is utilized in the hypothesis test. Table 7 lists the p values between the results of MOABC/D-MSA and results of the peer algorithms. The significance level is set at 0.05 in this study. If the p value is less than 0.05, it indicates that the result of MOABC/D-MSA is significantly

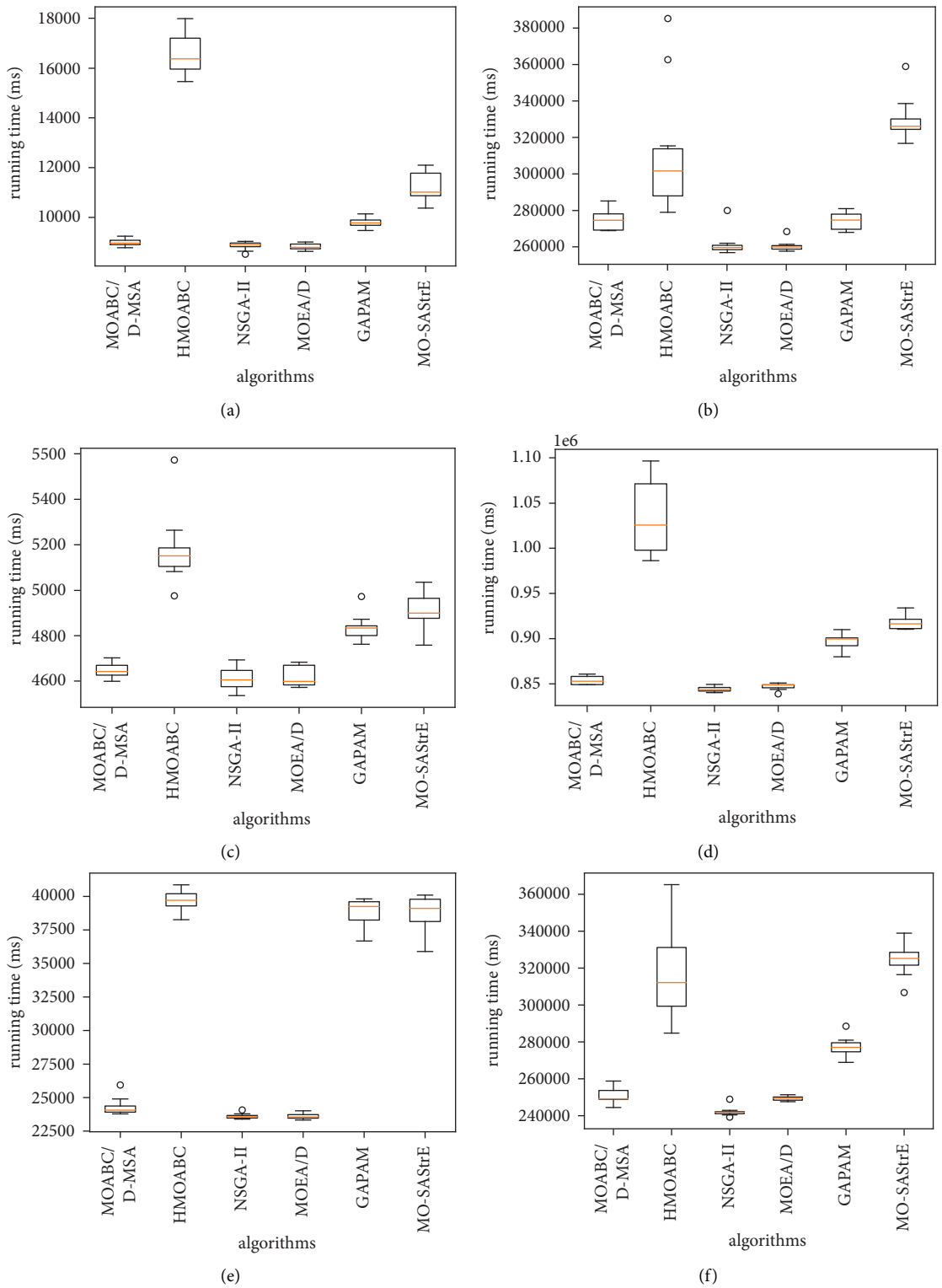

Figure 3: Box plots of running time. (a) BB11001. (b) BB12001. (c) BB20001. (d) BB30001. (e) BB40001. (f) BB50001.

better than the result of the competitor. According to Table 7, except for BB40038, BB50010, and BB50016, MOABC/D-MSA outperforms all its competitors in the test cases. Experimental results show that the results of the proposed algorithm are significantly better on most of the test cases. On BB40038, although the results of MOABC/D-MSA cannot significantly outperform NSGA-II and MOEA/D, according to Table 5, its results outperform the two algorithms in the aspect of %TC, %nonGap, and STRIKE. On BB50010, there is no significant difference between the

Table 7: Comparison on BAliBASE test cases.

Test case	HMOABC	NSGA-II	MOEA/D	GAPAM	MO-SAStrE
BB11001	0.005	0.005	0.005	0.005	0.005
BB11005	0.005	0.005	0.005	0.005	0.005
BB11018	0.005	0.005	0.005	0.004	0.05
BB11020	0.005	0.005	0.005	0.05	0.05
BB12001	0.05	0.005	0.05	0.05	0.05
BB12013	0.005	0.005	0.005	0.005	0.005
BB12022	0.005	0.003	0.004	0.005	0.005
BB12035	0.005	0.005	0.005	0.005	0.005
BB12044	0.005	0.005	0.005	0.005	0.005
BB20001	0.005	0.01	0.05	0.005	0.005
BB20010	0.005	0.05	0.01	0.01	0.01
BB20022	0.005	0.01	0.01	0.005	0.005
BB20033	0.005	0.05	0.01	0.005	0.005
BB20041	0.005	0.05	0.05	0.005	0.005
BB30001	0.005	0.005	0.01	0.005	0.005
BB30008	0.005	0.005	0.005	0.005	0.005
BB30015	0.005	0.01	0.01	0.005	0.005
BB30022	0.005	0.01	0.01	0.005	0.005
BB40001	0.005	0.05	0.05	0.005	0.005
BB40013	0.005	0.01	0.01	0.01	0.01
BB40025	0.01	0.05	0.0 5	0.005	0.01
BB40038	0.005	0.01	0.01	0.005	0.005
BB40048	0.01	0.01	0.10	0.01	0.01
BB50001	0.005	0.005	0.01	0.005	0.005
BB50005	0.005	0.01	0.05	0.005	0.005
BB50010	0.005	0.10	0.10	0.01	0.01
BB50016	0.005	0.05	0.10	0.005	0.01

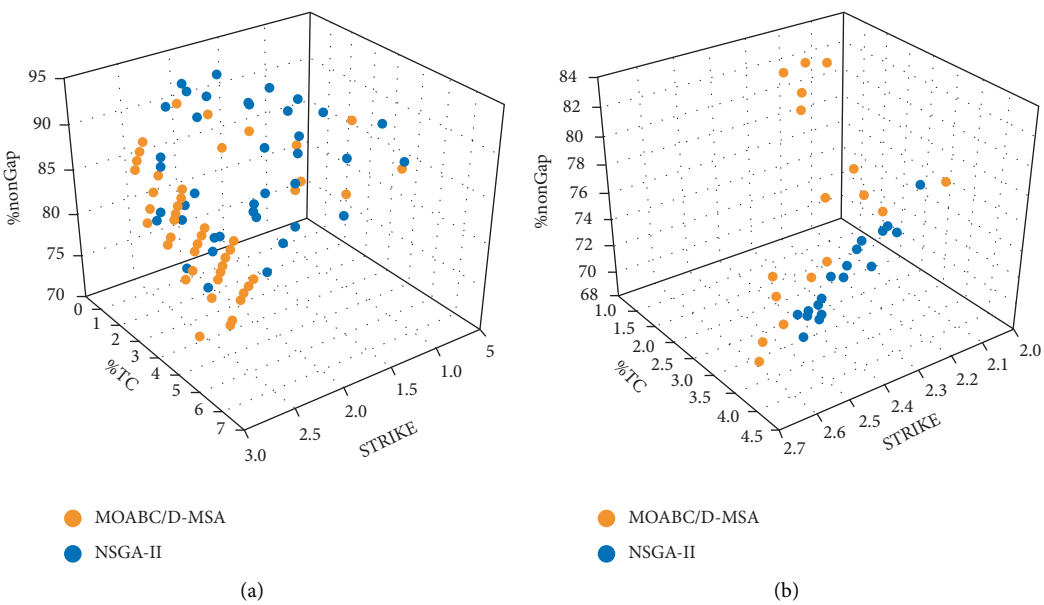

Figure 4: Comparison of the distribution of solutions of MOABC/D-MSA and NSGA-II. (a) BB11001. (b) BB12001.

results of MOABC/D-MSA and the results of NSGA-II, MOEA/D, GAPAM, and MO-SAStrE. On BB50016, although the results of MOABC/D-MSA cannot significantly outperform MOEA/D and MO-SAStrE, according to Table 6, its results outperform the two algorithms in the aspect of %TC, %nonGap, and STRIKE.

4.5. Solution Distribution. The statistical results show that nondominated solutions of MOABC/D-MSA can obtain the optimal results on all three objectives. Since the objectives cannot be optimized simultaneously, the results indicate that the distribution of results of MOABC/D-MSA is more uniform than the results of compared algorithms. Figure 4

illustrates the distribution of nondominated solution sets of MOABC/D-MSA and NSGA-II on BB11001 and BB12001, respectively. The figures show that solutions of MOABC/D-MSA have more comprehensive coverage of the PF. In this case, MOABC/D-MSA can provide solutions that meet different demands of users and researchers.

4.6. Discussion. Experimental results show that MOABC/D-MSA can obtain a better solution on all objectives when optimizing the three-dimensional multiobjective MSA problem. This result means that MOABC/D-MSA not only obtains solutions that are close to the PF of the problems but also obtains uniform solution distribution. In other words, MOABC/D-MSA achieves a good balance between convergence and diversity of solutions.

Meanwhile, MOABC/D-MSA is an efficient algorithm. For the same evaluation times, the running time of MOABC/D-MSA is competitive among the tested algorithms. This performance might be due to the fact that the proposed algorithm does not introduce additional calculations and complex search steps. Efficiency is important for the metaheuristics. In addition, MOABC/D-MSA avoids local optimum and controls the randomness during the research. An efficient algorithm could execute more iterations at the same time; therefore, it is more likely to obtain high-quality solutions.

Since the decomposition-based strategy is adopted in the employed bee phase, MOABC/D-MSA controls the uniform distribution of solutions. This strategy maximizes the difference between understandings while ensuring the quality of the nondominated solutions. In this way, the algorithm can provide more solutions that meet the different demands of algorithm users.

5. Conclusion

This study proposes MOABC/D-MSA, a decomposition-based artificial bee colony optimization algorithm for solving the MSA problem. The searching behavior of the employed bees is based on the PBI method, a decomposition-based strategy. The proposed algorithm considers both the convergence performance and the distribution of the alignments. The employed bees not only make the food sources converge to the PF but also ensure the distribution of the food sources, reflecting the real shape of the PF. The onlooker bees of MOABC/D-MSA accelerate the converging of food sources by utilizing high-quality solutions. MOABC/D-MSA uses the scout bee to get out of the local optimum. This study implements a comparative study on BAliBASE 3.0. The experimental results verify that MOABC/D-MSA has competitive convergence performance. Furthermore, nondominated solutions generated by MOABC/D-MSA show better performance in reflecting the PF the MSA problems. The superior distribution performance provides stronger decision support for biological researchers. For future studies, improving efficiency is a tough task and promising research direction.

Data Availability

There are no applicable datasets.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This work was supported by the Key Project of National Natural Science Foundation of China (U1908212) and the Startup Research Foundation for PhD of Liaoning Province (2020-BS-152).

References

[1] M. A. Nayeem, M. S. Bayzid, A. H. Rahman, R. Shahriyar, and M. S. Rahman, "Multiobjective formulation of multiple sequence alignment for phylogeny inference," *IEEE Transactions on Cybernetics*, 2020.

[2] H. Zhu, Z. He, and Y. Jia, "A novel approach to multiple sequence alignment using multiobjective evolutionary algorithm based on decomposition," *IEEE journal of biomedical and health informatics*, vol. 20, no. 2, pp. 717–727, 2015.

[3] R. K. Ramakrishnan, J. Singh, and M. R. Blanchette, "A reinforcement learning approach for multiple sequence alignment," in *Proceedings of the 2018 IEEE Eighteenth International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 61–66, IEEE, Taichung, Taiwan, October 2018.

[4] Á. Rubio-Largo, M. A. Vega-Rodríguez, and D. L. González-Álvarez, "A hybrid multiobjective memetic metaheuristic for multiple sequence alignment," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 499–514, 2015.

[5] Á. Rubio-Largo, L. Vanneschi, M. Castelli, and M. A. Vega-Rodríguez, "A characteristic-based framework for multiple sequence aligners," *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 41–51, 2016.

[6] K. Giannakis, C. Papalitsas, G. Theocharopoulou, S. Fanarioti, and T. Andronikos, "A quantum-inspired optimization heuristic for the multiple sequence alignment problem in biocomputing," in *Proceedings of the 2019 Tenth International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–8, IEEE, Patras, Greece, July 2019.

[7] Q. Zhan, N. Wang, S. Jin, R. Tan, Q. Jiang, and Y. Wang, "Probpfp: a multiple sequence alignment algorithm combining partition function and hidden Markov model with particle swarm optimization," in *Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1290–1295, IEEE, Madrid, Spain, December 2018.

[8] P. T. Doan and A. Takasu, "Sparse regression-based multiple sequence alignment," in *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1372–1377, IEEE, Shanghai, China, July 2019.

[9] N. Altwaijry, M. Almasoud, A. Almalki, and I. Al-Turaiki, "Multiple sequence alignment using a multiobjective artificial bee colony algorithm," in *Proceedings of the 2020 Third International Conference on Computer Applications & Information Security (ICCAIS)*, pp. 1–6, IEEE, Riyadh, Saudi Arabia, March 2020.

[10] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design,"

IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 11, pp. 6723–6742, 2020.

[11] L. Ma, M. Huang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, 2021.

[12] L. Ma, X. Wang, X. Wang, L. Wang, Y. Shi, and M. T. Huang, "Truthful combinatorial double auctions for mobile edge computing in industrial internet of things," *IEEE Transactions on Mobile Computing*, 2021.

[13] H. Zhao, C. Zhang, X. Zheng, C. Zhang, and B. Zhang, "A decomposition-based many-objective ant colony optimization algorithm with adaptive solution construction and selection approaches," *Swarm and Evolutionary Computation*, vol. 68, Article ID 100977, 2022.

[14] H. Zhao and C. Zhang, "A decomposition-based many-objective artificial bee colony algorithm with reinforcement learning," *Applied Soft Computing*, vol. 86, Article ID 105879, 2020.

[15] X. Lei, J. Sun, X. Xu, and L. Guo, "Artificial bee colony algorithm for solving multiple sequence alignment," in *Proceedings of the 2010 IEEE fifth international conference on bio-inspired computing: theories and applications (BIC-TA)*, pp. 337–342, IEEE, Changsha, China, September 2010.

[16] S. K. Burley, H. M. Berman, G. J. Kleywegt, J. L. Markley, H. Nakamura, and S. Velankar, "Protein data bank (pdb): the single global macromolecular structure archive," *Methods in Molecular Biology*, pp. 627–641, 2017.

[17] R. C. Edgar, "Muscle: multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Research*, vol. 32, no. 5, pp. 1792–1797, 2004.

[18] Q. Zhang and H. Li, "Moea/d: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[20] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part i: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.

[21] J. D. Thompson, P. Koehl, R. Ripp, and O. Poch, "Balibase 3.0: latest developments of the multiple sequence alignment benchmark," *Proteins*, vol. 61, no. 1, pp. 127–136, 2005.

[22] O. Kaiwartya, S. Prakash, and A. N. Hassan, "Multiple sequence alignment using genetic algorithm and non-dominant sorting genetic algorithm-ii (nsga ii) and variants," *Journal of Bioinformatics and Intelligent Control*, vol. 3, no. 4, pp. 294–299, 2014.

[23] F. Naznin, R. Sarker, and D. Essam, "Progressive alignment method using genetic algorithm for multiple sequence alignment," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 615–631, 2012.

[24] F. M. Ortuño, O. Valenzuela, F. Rojas et al., "Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: structural information, non-gaps percentage and totally conserved columns," *Bioinformatics*, vol. 29, no. 17, pp. 2112–2121, 2013.

[25] Á. Rubio-Largo, M. A. Vega-Rodríguez, and D. L. González-Álvarez, "Hybrid multiobjective artificial bee colony for multiple sequence alignment," *Applied Soft Computing*, vol. 41, pp. 157–168, 2016.

[26] S. M. Taheri and G. Hesamian, "A generalization of the wilcoxon signed-rank test and its applications," *Statistical Papers*, vol. 54, no. 2, pp. 457–470, 2013.

[27] Y. Sun, G. G. Yen, and Z. Yi, "Igd indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 173–187, 2018.