A Multi-objective Artificial Bee Colony Algorithm for Multiple Sequence Alignment

Ying Yu[1], Chen Zhang[1(✉)], Lei Ye[2], Ming Yang[3], and Changsheng Zhang[1]

[1] Software College of Northeastern University, Shenyang 110169, China
[2] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310000, China
[3] School of Information Network Security, People's Public Security University of China, Beijing 10038, China

Abstract. The multiple sequence alignment (MSA) problem is essential in biological research for finding specific relationship between the biologic sequences and their functions. This paper proposes a multi-objective artificial bee colony optimization algorithm for MSA (MOABC-MSA), which uses three kinds of searching to optimize a multi-objective MSA problem. The employed bee searching aims to make the solutions converge to the Pareto front (PF) of the problem; the onlooker bee accelerates the convergence speed; the scout bee facilitates the algorithm to avoid the local optimal. A comparative experiment is implemented on BAliBASE 3.0, a MSA benchmark. Experimental results show that the proposed algorithm has competitive performance with state-of-the-art metaheuristic algorithms.

Keywords: Multiple sequence alignment · Multi-objective optimization · Artificial bee colony optimization

1 Introduction

The multiple sequence alignment (MSA) problem aims to align three or more sequences simultaneously. MSA can help researchers to identify regions that exists same elements, which might contain valuable information.

The MSA problem has been proven to be a NP-complete problem with the sum-of-pairs (SP) metric [1], and NP-hard for most of existing metrics [2]. Different kinds of algorithms are developed to solve MSA problem.

Literature [3] finds six groups of approaches for MSA problem: (1) exact methods; (2) progressive methods; (3) consistency-based methods; (4) iterative methods; (5) evolutionary algorithms; (6) structure-based methods.

The mataheuristic algorithms have shown competitive performance in optimizing MSA problems. A hybrid multiobjective metaheuristic combining shuffled

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2022
Published by Springer Nature Switzerland AG 2022. All Rights Reserved
D. Jiang and H. Song (Eds.): SIMUtools 2021, LNICST 424, pp. 564–576, 2022.
https://doi.org/10.1007/978-3-030-97124-3_44

frog-leaping algorithm with Kalgin algorithm is proposed in [3]. Literature [4] proposed a characteristic-based framework for MSA, which extracts the characteristics of input unaligned sequences and aligns the sequences with specific configuration according to the characteristics. A bi-objective evolutionary algorithm with a tree-based initialization method and a gap-re-inserting-based mutation operator is proposed in [1]. Literature [5] proposes a quantum-inspired heuristic optimization method for MSA. A multi-objective formulation for MSA is proposed in [9] for inferring the evolutionary history relating. ProbPFP [6] combines the partition function and the hidden Markov model (HMM), where the parameters is optimized by a particle swarm optimization (PSO). Literature [7] proposes an algorithm that employs sparse approximation to reduce computational cost for the relaxation. Literature [8] proposes a hybrid artificial bee colony optimization (ABC) algorithm for MSA, which performs a single-point crossover for employed bee phase and a multiple mutation operator that contains four kinds of mutations operators for onlooker bee phase.

Early literatures evaluates the alignments by one score function. However, when optimizing the MSA problem, researchers have more than one requirement for aligned sequences. To meet different requirements of researchers, the MSA problem is designed to contain multiple optimization objectives. [3] adopts the weighted sum-of-pairs function with affine gap penalties (WSP) and the number of total conserved columns (TC) score. [4] uses the Q-score (i.e., sum-of-pairs (SP) score) and total column (TC) score as the objective functions. In [1], the MSA problem is a bi-objective minimization problem, where the first objective function is the scoring function that minimizes the number of gaps, the second objective function is the opposite of SP function. Four objectives are proposed in [9]: non-gap columns for the calculation of entropy, the similarity of columns containing one or more gaps, the similarity of column containing no gap, and the number of consecutive gaps. [7] develops a relaxed formulation for the MSA problem based on a regression-coding framework.

This paper proposes a novel artificial bee colony optimization algorithm for MSA (MOABC-MSA). The employed bee stage achieves optimization for each sub-problem. The onlooker bees stage is based on a non-dominated ranking-based roulette wheel selection, which can obtain high-quality solutions with high diversity. The scout bee works could facilitate the algorithm in avoiding local optimums. The details of the MOABC-MSA is introduced in the third section of this paper.

The remainder parts of this paper is organized as follows. The second section introduces the definition of the multi-objective MSA problem. Section 3 describes the design and implementation of the proposed MOABC-MSA. Section 4 compares the MOABC-MSA with state-of-the-art metaheuristics on 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

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 maximizing the accuracy of the alignment. Maximizing %TC ensures there are more columns that the residues are exactly the same, i.e., 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 shown in Eq. 1:

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

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 [10].

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:

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

where f_{ij} is the frequency of contacts involving i and j across all observed *residue-residue* contacts, 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 Eq. 3:

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

where S_l is the l^{th} column of S, $S_l = s_{il} \forall i = 1, ..., k$, and $totalColumn(S_l)$ is defined following Eq. 4:

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

%nonGap measures the number of residues with respect to the number of gaps into the alignment. This objective function is shown in Eq. 5:

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

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 Eq. 6:

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

3 MOABC-MSA

3.1 Representation of Individuals

This paper adopted a 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. For example, if a sequence is {REDH-PDLIQ—NAK-K–}, it is represented as: {(5,5), (11,14), (18,18), (20,21)}.

3.2 Crossover and Mutation Operators

This paper employs a single-point crossover operator [3]. There are three kinds of mutation operators adopted in the proposed algorithm: shift-closed gaps, non-gap group splitting, and adjacent gap groups merging [16].

3.3 Algorithm Overview

Algorithm 1 shows the framework of the proposed MOABC-MSA. First, the algorithm uses MUSCLE, a non-metaheuristic method, to initialize N random food sources that represents N candidate alignments. Meanwhile, an archive NA for storing non-dominated 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 behaviours of two kinds of bees until it meets the stop criterion. Finally, the algorithm outputs non-dominated solutions and their corresponding scores.

3.4 Employee

Algorithm 2 shows the execution process of the employed bee stage. Each employed bee is assigned a food source. Each bee performs single-point crossover operation. For the i^{th} bee, the first individual of the crossover operation is the i^{th} food source, the second individual is randomly selected. After the crossover, two new candidate alignments are generated. Then the proposed algorithm performs shift-closed gaps mutation operator on the newly-generated candidate alignments. The new solutions are evaluated by the objective functions and merged into the candidate solution set.

The non-dominated sorting-based selection [13] is adopted in employed bee stage. The candidate solutions in each iteration is divided into several levels. The first level is non-dominated solutions among all candidates. The second is

Algorithm 1: Algorithm framework of MOABC-MSA

```
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
    end
    update non-dominated set NA;
end
output non-dominated solutions;
```

non-dominated solutions among candidates except the first level. The rest of the food sources is sorted in the same manner. If the algorithm can obtain N optimal solutions by select certain levels of candidate solutions into the population of next generation, it goes into the next iteration; if not, assuming that the first s levels are selected, the algorithm associates solutions in the $(s+1)^{th}$ level with N uniformly distributed reference vectors (RVs), and choose $N-\sum_{i=0}^{s}|level_i|$ solutions into the population of next generation by niche-preservation method [13].

3.5 Onlooker

The onlooker bees prefer to exploit high-quality solutions founded by employed bees. Meanwhile, the bees should maintain diversity of food sources.

At this stage, each onlooker bees select food sources to search according to the solution quality. Inspired by the fast non-dominated sort method [12], the proposed algorithm divides the food sources into several ranks.

When selecting food sources, an onlooker bee first select a rank based on roulette wheel selection, which guarantees that the non-dominated food sources are more likely to be selected. Assuming that there are M ranks among the food sources, the selected probability of the i^{th} rank is calculated according to Eq. 7:

$$p_i = \frac{M-i+1}{\sum_{j=1}^{M} j} \tag{7}$$

Next, the onlooker bee selects a food source randomly from the selected rank.

Each onlooker bee performs a single-point crossover operation. Then it performs the shift-closed gaps mutation, non-gap group splitting mutation, and adjacent gap groups merging mutation in sequence.

The food sources are updated by the non-dominated and crowded-based sorting selection [12]. The onlooker bee stage is described in Algorithm 3.

Algorithm 2: Searching behaviour of employed bees

Input: N food sources;
Output: N new food sources;
initialize $\boldsymbol{\omega}_1$ to $\boldsymbol{\omega}_N$;
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
  evaluate new solutions;
  merge food sources and new solutions;
  non-dominated sorting;
  if $N = \sum_{i=0}^{s} |level_i|$ **then**
    select s levels as new food source;
  end
  else
    select new food source using sorted levels and niche-perservation method;
  end
  update food sources;
end

Algorithm 3: Searching behaviour of onlooker bees

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
  evaluate new solutions;
  merge food sources and new solutions;
  perform non-dominated and crowded-based sorting selection;
  update food sources;
end

3.6 Scout Bee Phase

The scout bees work when the optimal solution of a sub-problem is not updated for more than k iterations. The scout bees performs the same crossover operator and mutation operator with the onlooker bees. Algorithm 4 shows the searching behaviour of scout bee.

Algorithm 4: Searching behaviour of scout bee

```
   Input: sequences to be aligned;
   Output: aligned sequences;
 1 initialize N random food sources;
 2 initialize non-dominated set NA;
 3 evaluate food sources;
 4 while termination criterion is false do
 5 |   for each sub-problem do
 6 |   |   if the optimal solution is not updated for more than k iterations then
 7 |   |   |   perform single-point crossover operation;
 8 |   |   |   perform shift-closed gaps mutation operation;
 9 |   |   |   perform non-gap group splitting;
10 |   |   |   perform djacent gap groups merging;
11 |   |   end
12 |   end
13 |   evaluate new solutions;
14 |   merge food sources and new solutions;
15 |   find optimal solution of each sub-problem;
16 |   update food sources;
17 end
```

4 Experimental Result

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

This paper uses the 3.0 version of Benchmark alignment database (BAliBASE) to test the proposed algorithm. There are 218 sets of sequences in BAliBASE 3.0, they are divided into six families: RV11, RV12, RV20, RV30, RV40, and RV50.

This experiment selects twenty-seven test cases from BAliBASE 3.0 to implement the comparative study between the proposed MOABC-MSA and other genetic and metaheuristic algorithms. The selected instances are: BB11001, BB11005, BB11018, BB11020 from RV11, BB12001, BB12013, BB12022, BB12035, BB12044 from RV12, BB20001, BB20010, BB20022, BB20033, BB20041 from RV20, BB30001, BB30008, BB30015, BB30022 from RV30, BB40001, BB40013, BB40025, BB40038, BB40048 from RV40, BB50001, BB50005, BB50010, BB50016 from RV50.

This paper compares MOABC-MSA with several state-of-the-art evolutionary or metaheuristic algorithms. The competitor algorithms includes: NSGA-II [18], MOEA/D [17], GAPAM [15], MO-SAStrE [16], and HMOABC [19]. The parameters of the peer algorithms are set according to their original literatures.

Table 1. Comparison on RV11

Test case		MOABC-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

Table 2. Comparison on RV12

Test case		MOABC-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

4.1 Statistic 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 the randomness. The termination criterion is set as 25,000 times of evaluation to guarantee the fairness of the experiment. For MOABC-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.

Table 3. Comparison on RV20

Test case		MOABC-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

Table 4. Comparison on RV30

Test case		MOABC-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

Tables 1, 2, 3, 4, 5 and 6 shows the best value of STRIKE, %TC, and %nonGap among solutions found by the tested algorithms for each test case. The bold number indicates the optimum. Table 1, 2 and 3 shows that for instances in RV11, RV12, and RV20, MOABC-MSA obtains the optimal STRIKE, %TC, and %nonGap on all test cases. For BB30008, both MOABC-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-MSA outperforms the compared algorithms in both the three objectives.

Table 5. Comparison on RV40

Test case		MOABC-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

Table 6. Comparison on RV50

Test case		MOABC-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

4.2 Hypothesis Results

This paper uses the Wilcoxon signed-rank hypothesis test to investigate the difference between the performance of the MOABC-MSA and results of the competitors. The objective values of each solution is normalized to real number between zero and one, then the normalized solutions are evaluated by the IGD indicator [14]. The IGD indicator works out a scalarized score for each non-dominated solution set. Finally, the IGD value of each solution set is utilized in the hypothesis test. Table 7 shows the p-values between results of MOABC-MSA

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

and results of the peer algorithms. The significance level is set at 0.05 in this paper. If the p-value is less than 0.05, it indicates that the result of MOABC-MSA is significantly better than the result of the competitor. According to Table 7, except for BB40038, BB50010, and BB50016, MOABC-MSA outperforms all its competitors in the test cases.

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

4.3 Further Analysis

The computation complexity of the employee phase and the onlooker phase is $O(N^2)$. Therefore, the computation complexity of the MOABC-MSA is $O(N^2)$.

Compared to other heuristic multi-objective algorithms, the proposed algorithm uses three kinds of searching strategies alternately. Therefore, MOABC-MSA can balance the convergence and diversity during the searching process. Different from existing ABC-based algorithms, MOABC-MSA uses a roulette wheel selection according to the performance in Pareto dominance relationship. The selection strategy uses the information of non-dominated solutions to accelerate the convergence to the PF.

5 Conclusion

This paper proposes MOABC-MSA, an artificial bee colony optimization algorithm for solving MSA problem. 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 can reflect the real shape of the PF. The onlooker bees of MOABC-MSA accelerates the converging of food sources by utilizing high-quality solutions. MOABC-MSA uses the scout bee to avoid the local optimum. The comparative study on BAliBASE 3.0 verifies that MOABC-MSA has competitive performance. For the future study, improving the efficiency is a tough task and promising research direction.

References

1. Zhu, H., He, Z., Jia, Y.: A novel approach to multiple sequence alignment using multiobjective evolutionary algorithm based on decomposition. IEEE J. Biomed. Health Inform. **20**(2), 717–727 (2015)
2. Ramakrishnan, R.K., Singh, J., Blanchette, M.: RLALIGN: a reinforcement learning approach for multiple sequence alignment. In: 2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE), pp. 61–66. IEEE (2018)
3. Rubio-Largo, Á., Vega-Rodríguez, M.A., González-Álvarez, D.L.: A hybrid multiobjective memetic metaheuristic for multiple sequence alignment. IEEE Trans. Evol. Compu. **20**(4), 499–514 (2015)
4. Rubio-Largo, Á., Vanneschi, L., Castelli, M., Vega-Rodríguez, M.A.: A characteristic-based framework for multiple sequence aligners. IEEE Trans. Cybern. **48**(1), 41–51 (2016)
5. Giannakis, K., Papalitsas, C., Theocharopoulou, G., Fanarioti, S., Andronikos, T.: A quantum-inspired optimization heuristic for the multiple sequence alignment problem in bio-computing. In: 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1–8. IEEE (2019)
6. Zhan, Q., Wang, N., Jin, S., Tan, R., Jiang, Q., Wang, Y.: ProbPFP: a multiple sequence alignment algorithm combining partition function and hidden Markov model with particle swarm optimization. In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1290–1295. IEEE (2018)

7. Doan, P.T., Takasu, A.: Sparse regression-based multiple sequence alignment. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 1372–1377. IEEE (2019)
8. Altwaijry, N., Almasoud, M., Almalki, A., Al-Turaiki, I.: Multiple sequence alignment using a multiobjective artificial bee colony algorithm. In: 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), pp. 1–6. IEEE (2020)
9. Nayeem, M.A., Bayzid, Md.S., Rahman, A.H., Shahriyar, R., Rahman, M.S.: Multiobjective formulation of multiple sequence alignment for phylogeny inference. IEEE Trans. Cybern. (2020)
10. Burley, S.K., Berman, H.M., Kleywegt, G.J., Markley, J.L., Nakamura, H., Velankar, S.: Protein Data Bank (PDB): the single global macromolecular structure archive. In: Wlodawer, A., Dauter, Z., Jaskolski, M. (eds.) Protein Crystallography. MMB, vol. 1607, pp. 627–641. Springer, New York (2017). https://doi.org/10.1007/978-1-4939-7000-1_26
11. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2008)
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.A.M.T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
13. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014)
14. Sun, Y., Yen, G.G., Yi, Z.: IGD indicator-based evolutionary algorithm for many-objective optimization problems. EEE Trans. Evol. Comput. **23**(2), 173–187 (2018)
15. Naznin, F., Sarker, R., Essam, D.: Progressive alignment method using genetic algorithm for multiple sequence alignment. IEEE Trans. Evol. Comput. **16**(5), 615–631 (2012)
16. Ortuño, F.M., et al.: Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: structural information, non-gaps percentage and totally conserved columns. Bioinformatics **29**(17), 2112–2121 (2013)
17. Huazheng, Z., He, Z., Jia, Y.: A novel approach to multiple sequence alignment using multiobjective evolutionary algorithm based on decomposition. IEEE J. Biomed. Health Inform. **20**(2), 717–727 (2016)
18. Kaiwartya, O., et al.: Multiple sequence alignment using genetic algorithm and Non-Dominant Sorting Genetic Algorithm-II (NSGA II) and variants. J. Bioinform. Intell. Control **3**(4), 294–299 (2014)
19. Zhang, H., Zhu, Y., Zou, W., Yan, X.: A hybrid multi-objective artificial bee colony algorithm for burdening optimization of copper strip production. Appl. Math. Model. **36**(6), 2578–2591 (2012)