A Multi-strategy Artificial Bee Colony Algorithm with Neighborhood Search

Can Sun, Xinyu Zhou(✉), and Mingwen Wang

School of Computer and Information Engineering,
Jiangxi Normal University, Nanchang 330022, China
xyzhou@jxnu.edu.cn

Abstract. As an effective swarm intelligence based optimization technique, artificial bee colony (ABC) algorithm has become popular in recent years. However, its performance is still not satisfied in solving some complex optimization problems. The main reason is that both of the employed bee phase and onlooker bee phase use the same solution search equation to generate new candidate solutions, and the solution search equation is good at exploration but poor at exploitation. To solve this problem, in this paper, we propose a multi-strategy artificial bee colony algorithm with neighborhood search (MSABC-NS). In MSABC-NS, a multi-strategy mechanism is designed to use two different solution search equations, and a neighborhood search mechanism is introduced to make full use of good solutions. Experiments are conducted on 22 widely used benchmark functions, and three different ABC variants are included in the comparison. The results show that our approach can achieve better performance on most of the benchmark functions.

Keywords: Artificial bee colony · Exploration and exploitation · Multi-strategy mechanism · Neighborhood search

1 Introduction

Artificial bee colony (ABC) algorithm is a swarm intelligence based optimization technique, and it has become popular in the community of evolutionary algorithms (EAs) in recent years. Some previous works have pointed out that the performance of ABC is competitive in comparison with other EAs, such as genetic algorithm (GA) [1, 2], particle swarm optimization (PSO) [3, 4], and differential evolution (DE) [5]. The basic ABC is proposed by Karaboga in 2005, which simulates the honeybee's foraging behavior [6, 7]. The basic ABC contains three different kinds of bees, i.e., employed bees, onlooker bees, and scout bees. These three different kinds of bees have different tasks, but they cooperate to maximize the nectar amount, which implies that they cooperate to search the optimal solution of optimization problems.

Although the ABC algorithm has shown good performance, its performance is still not satisfactory in solving some complex optimization problems. Some researchers have indicated that the basic ABC tends to show slow convergence speed. The main reasons possibly include two aspects. On the one hand, the solution search equation in the basic ABC algorithm is good at exploration but poor at exploitation, which leads to an improper balance between the exploration and exploitation capabilities. As we know, a

Y. Tan et al. (Eds.): ICSI 2019, LNCS 11655, pp. 310–319, 2019.
https://doi.org/10.1007/978-3-030-26369-0_29

good balance between the exploration and exploitation capabilities is the key issue for the performance of EAs. On the other hand, different kinds of bees have different tasks, so the responsibilities should also be different. In the basic ABC, however, the same solution search equation is used by both of the employed bees and onlooker bees to generate new candidate solutions, so the performance is limited in this context.

To solve these problems, many different improved ABC variants have been proposed in recent years, and most of them focus on designing new solution search equations. For instances, Zhu et al. [8] proposed a modified ABC variant (GABC) in which the global best solution is integrated into the solution search equation to improve the exploitation capability. Based on the idea of using good solutions, Gao et al. [9] modified the original solution search equation inspired by the DE mutation strategy DE/best/1 in their proposed MABC algorithm. In contrast with the two aforementioned works of using the global best solution, Karaboga et al. [10] designed a neighborhood search mechanism based on the Euclidean distance in their proposed qABC algorithm, which aims to improve the exploitation capability while without losing diversity. Very recently, Cui et al. [11] proposed a modified version of GABC, called MPGABC, in which a multi-strategy of using different solution search equations is designed, and the reported experimental results show the effectiveness and efficiency of MPGABC.

Inspired by the above ABC variants, in this paper, we propose a multi-strategy ABC with neighborhood search (MSABC-NS). As above mentioned, the possible reasons of the deficiencies include two aspects, so we make two corresponding modifications to deal with the deficiencies, respectively. First, we design a multi-strategy mechanism in which two different solution search equations are used through a simple IF-ELSE structure. Second, we introduce a neighborhood search mechanism to make full use of good solutions. It's necessary to point out that our modifications is based on the structure of MPGABC, and the proposed MSABC-NS algorithm can be considered as an improved version of MPGABC to some extent. In the experiments, a suite of widely used 22 benchmark functions is employed to estimate the performance of our approach, and three different ABC variants are included in the comparison. The experimental results show that our approach can achieve better performance on most of the benchmark functions.

The remainder of this paper is organized as follows. We will briefly introduce the basic ABC algorithm in the Sect. 2, while our proposed algorithm will be described in detail in the Sect. 3. The Sect. 4 will show the experiments and the corresponding analysis. The summary of this paper is given in the last section.

2 The Basic ABC Algorithm

The optimization process of the basic ABC algorithm includes four phases, i.e., the initialization phase, the employed bee phase, the onlooker bee phase and the scout bee phase. After initialization phase, ABC turns into a loop of the employed bee phase, the onlooker bee phase and the scout bee phase until the termination condition is met. These three phases have different responsibilities in terms of their roles in the optimization process. In the employed bee phase, the employed bees are responsible for exploration, while the onlooker bees has the responsibility of exploitation in the onlooker bee phase, and the scout bees discards food source which cannot be exploited further. These four phases are described in detail as follows.

(1) The initialization phase

In the initialization phase, the initial food sources are generated randomly according to the Eq. (1). It's worth noting that a food source represents a candidate solution of the optimization problem.

$$x_{i,j} = x_j^{min} + rand(0,1) \cdot \left(x_j^{max} - x_j^{min}\right), \tag{1}$$

where x_i represents the ith food source, x_j^{max} and x_j^{min} represent the boundary of the jth dimension.

(2) The employed bee phase

In the employed bee phase, each employed bee searches a new food source by using the solution search equation listed in the Eq. (2). After all of the employed bees finish their search, they will share the relevant information with the onlooker bees which include the nectar amount and the positions of the food sources.

$$v_{i,j} = x_{i,j} + \emptyset_{i,j} \cdot \left(x_{k,j} - x_{i,j}\right), \tag{2}$$

where x_i represents the current food source or parent candidate solution, v_i indicates the new candidate solution. $\emptyset_{i,j}$ is an uniform random number within the range $[-1,1]$. $j \in \{1,2\ldots,D\}$ is a randomly selected dimension, and D is the dimensionality of the optimization problem. If v_i is better than x_i in terms of the fitness value, then x_i will be replaced with v_i in the next generation. It's worth noting that the fitness value is calculated by using the following Eq. (3), where fit_i and $f\,(x_i)$ is the fitness value and the objective value of x_i respectively.

$$fit_i = \begin{cases} \frac{1}{1+f(x_i)} \\ 1 + abs(f(x_i)) \end{cases}. \tag{3}$$

(3) The onlooker bee phase

In the onlooker bee phase, onlooker bees will continue to search new food sources, but this is different from the employed bee phase in which each food source has the chance to be searched. Instead, onlooker bees favor searching good food sources based on the received information from the employed bees. The probability of food source whether is selected depends on the fitness value, and the following Eq. (4) is used to calculate the probability.

$$p_i = fit_i / \sum\nolimits_{j=1}^{NS} fit_j. \tag{4}$$

It can be seen that the bigger the value of fit_i is, the higher probability of being selected for the ith food source. After determining which food source should be selected, the onlooker bees will use the same solution equation listed in the Eq. (2) to generate new candidate solutions.

(4) The scout bee phase

In the scout bee phase, if a food source cannot be exploited for *limit* times, it is considered to be exhausted, and it will be discarded by the scout bee. As an alternative, this discarded food source will be randomly initialized by using the above Eq. (1).

3 The Proposed MSABC-NS Algorithm

3.1 The Multi-strategy Mechanism

In the basic ABC algorithm, both of the employed bee phase and the onlooker bee phase use the same solution search equation to generate new candidate solutions. Due to the strong exploration capability but weak exploitation capability of the solution search equation, the performance of the basic ABC is not satisfactory in solving some complex problems. Therefore, we attempt to design a multi-strategy ABC variant to solve this deficiency, which is beneficial to take the advantages of different strategies.

In fact, there already exist some works about multi-strategy ABC variant. Wang et al. [12] designed a multi-strategy mechanism in which a strategy pool is constructed based on three solution search strategies. When the food source cannot be successfully updated by some solution search strategy, a different new strategy will be randomly selected from the strategy pool to replace the old strategy. However, due to the reason of random selection, the efficiency of the algorithm can be improved further.

Very recently, Cui et al. [11] designed a new multi-strategy mechanism in their proposed MPGABC algorithm in which two different solution search strategies are included. The first strategy is the same with the basic solution search equation, while the second one uses the information of the global best solution to guide search. To control the frequency of these two strategies, a control parameter P is introduced which has significant impact on the performance of MPGABC. However, a fixed value of P is employed, which may hinder the versatility of MPGABC.

Being inspired by the above multi-strategy ABC variants, we attempt to propose a simple but effective multi-strategy mechanism to enhance the performance of ABC. In the algorithm 1, the structure of the proposed multi-strategy mechanism is first given for a clear description. In there, X_{pbest} is a randomly selected food source from a food source set which contains the top N best food sources. The value of N is set to $SN \times q$, SN is the number of food sources and q is a random number in the range $\left[\frac{2}{SN}, 0.1\right]$. $\varphi_{i,j}$ is an uniformly distributed random number in the range $[0, 1.5]$. X_k is a randomly selected food source from the population. $\emptyset_{i,j}$ is an uniformly distributed random number in the range $[-1, 1]$.

Algorithm 1: The proposed multi-strategy mechanism

1. $v_{i,j} = x_{i,j} + \varphi_{i,j} \cdot \left(x_{pbest,j} - x_{i,j}\right)$
2. **if** $fit(v_i) > fit(x_i)$
3. $\quad x_i = v_i$
4. **else**
5. $\quad v_{i,j} = x_{i,j} + \emptyset_{i,j} \cdot \left(x_{i,j} - x_{k,j}\right)$
6. **end**

It can be seen from the algorithm 1, X_{pbest} is a member of the top N food sources which implies it has relatively good fitness value, and it can be considered as an elite solution. What's more, $\varphi_{i,j}$ is set to be larger than 0, this is helpful to push X_i to move toward X_{pbest}, and the convergence rate of the algorithm can be speeded up to some extent. It's worth noting that the value of N is not fixed, which is beneficial to prevent the algorithm from being too greedy [13]. If the food source X_i cannot be improved by the X_{pbest}, the original solution search equation in the basic ABC is used as an alternative. To some extent, this IF-ELSE structure can enhance the exploitation capability while without losing the exploration capability. In addition, the new probability model in the MGPABC algorithm is still kept in our proposed algorithm for the onlooker bees.

3.2 The Neighborhood Search Mechanism

In order to further improve the performance of our approach, the global neighborhood search (GNS) operator proposed by Wang et al. [14] is introduced into our approach. The neighborhood search is to search the vicinity area of candidate solutions for better solutions. In fact, in different EAs, researchers have designed a variety of different types of neighborhood search operations. For example, Wang et al. [14] proposed a global neighborhood search operation to solve the problem of slow convergence speed of PSO algorithm. Although different neighborhood search operations have their own characteristics, they all effectively improve the performance of the corresponding algorithm. In this paper, we directly adopt the GNS operation to further improve the performance of our approach. The GNS operation has a simple structure but good performance, and the Eq. (5) is used in the GNS operation

$$TX_i = r_1 \cdot X_i + r_2 \cdot X_{gbest} + r_3 \cdot (X_a - X_b). \tag{5}$$

In the Eq. (5), TX_i is the candidate solution, X_{gbest} is the global best solution in the population, X_a and X_b are exclusive food sources and they are randomly selected from the population, what's more, they have to be different from X_i. The parameters r_1, r_2 and r_3 are exclusive random numbers within [0, 1], and they have to meet the condition $r_1 + r_2 + r_3 = 1$. A clear demonstration of the GNS operation is presented in Fig. 1. In order to control use frequency of the GNS operation, we set the probability of using the GNS operation to 0.1.

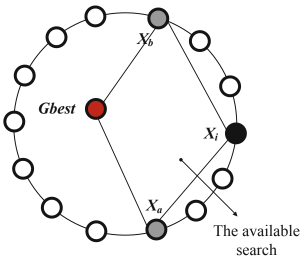

Fig. 1. The global neighborhood search operator.

3.3 Pseudo-code of MSABC-NS

In this paper, we propose an improved ABC variant (MSABC-NS) by combining a simple multi-strategy mechanism with the GNS operator. The pseudo-code of

MSABC-NS is described in the algorithm 2. In there, *FEs* is the number of used fitness function evaluations, and *MaxFEs*, as the stopping criterion, is the maximal number of fitness function evaluations. The control parameter *P* is set to 0.1, $trial_i$ represents the non-updated times of the i th food source.

Algorithm 2: Pseudo-code of MSABC-NS

Randomly generate SN food sources and calculate their fitness values, $FEs = SN$;
while $FEs \leq MaxFEs$ **do**
    /* Employed bee phase */
    Rank all the food sources according to their fitness value;
    for $i = 1$ to SN **do**
        Randomly select X_{pbest} and X_k from the population;
        Generate a new candidate solution V_i according to the algorithm 1;
        Calculate the fitness value of V_i, $FEs = FEs+1$;
        if $f(V_i) < f(X_i)$ **then**
            Replace X_i with V_i;
        end if
    end for
    /* Onlooker bee phase */
    Calculate the probability p_i according to the new probability model;
    for $i = 1$ to SN **do**
        Choose a food source X_j by the roulette wheel selection mechanism;
        Randomly select X_{pbest} and X_k from the population;
        Generate a new candidate solution V_j by algorithm 1;
        Calculate the fitness value of V_j, $FEs = FEs+1$;
        if $f(V_j) < f(X_j)$ **then**
            Replace X_j with V_j;
        end if
    end for
    /* Scout bee phase */
    if $\max(trial_i) > limit$ **then**
        Replace X_i with a new randomly generated food source according to the equation (1), $trial_i = 0$;
    end if
    /* The global neighborhood search operator */
    for $i = 1$ to SN **do**
        Generate a random number $rand \in [0,1]$;
        if $rand < P$ **then**
            Generate a trial solution TX_i according to the equation (5);
            Calculate the fitness value of TX_i, $FEs = FEs + 1$;
            if $f(TX_i) < f(X_i)$ **then**
                Replace X_i with TX_i;
            end if
        end if
    end for
end while

4 Experiments and Analysis

4.1 Benchmark Functions

We use 22 widely used benchmark functions to verify our approach. In these functions, F1–F9 are the unimodal functions, F6 is an uncontinuous step function, F10 is multimodal when its dimension is more than three, and F11–F22 are multimodal functions. The detailed definitions about the functions are listed in the Table 1. For the parameter settings of our approach, *SN* is set to 50, dimensionality of the functions *D* is set to 30, *maxFES* is set to $5000 \cdot D$ and *limit* is set to $SN \cdot D/2$. Each algorithm is run 25 times per function, the average function values are recorded.

Table 1. The 22 benchmark functions used in the experiments.

Function	Name	Range	Optimum
F1	Sphere	$[-100, 100]^D$	0
F2	Elliptic	$[-100, 100]^D$	0
F3	SumSquare	$[-10, 10]^D$	0
F4	SumPower	$[-1, 1]^D$	0
F5	Schwefel2.22	$[-10, 10]^D$	0
F6	Schwefel2.21	$[-100, 100]^D$	0
F7	Step	$[-100, 100]^D$	0
F8	Exponential	$[-10, 10]^D$	0
F9	Quartic	$[-1.28, 1.28]^D$	0
F10	Rosenbrock	$[-5, 10]^D$	0
F11	Rastrigin	$[-5.12, 5.12]^D$	0
F12	NCRastrigin	$[-5.12, 5.12]^D$	0
F13	Griewank	$[-600, 600]^D$	0
F14	Schwefel2.26	$[-500, 500]^D$	0
F15	Ackley	$[-50, 50]^D$	0
F16	Penalized1	$[-100, 100]^D$	0
F17	Penalized2	$[-100, 100]^D$	0
F18	Alpine	$[-10, 10]^D$	0
F19	Levy	$[-10, 10]^D$	0
F20	Weierstrass	$[-1, 1]^D$	0
F21	Himmelblau	$[-5, 5]^D$	−78.33236
F22	Michalewicz	$[0, \pi]^D$	−30, −50, −100

4.2 Verifications of the Proposed Algorithmic Components

Our approach includes two components, i.e., the multi-strategy mechanism and the GNS operator. To verify these two components, two compared algorithms are designed as baselines, i.e., MPGABC-IE and MPGABC-NS. MPGABC-IE represents the MPGABC algorithm only replace its multi-strategy mechanism with our proposed multi-strategy mechanism, while MPGABC-NS only adds the GNS operator. The compared results are shown in the Table 2 and the best results are shown in **boldface**.

As we can see from the Table 2, when compared with MPGABC, MPGABC-IE can get better results on the functions F1, F2, F3, F5, F10, F18 and F20, this implies that the proposed multi-strategy mechanism has shown better performance. Similarly, MSABC-NS has achieved better results on most test functions, and this indicates that the GNS operator indeed improve the performance of the algorithm. After by combing these two components, MSABC-NS has shown the best performance among the included four algorithms.

Table 2. Efficiency of the proposed algorithmic components.

Function	MPGABC	MPGABC-IE	MPGABC-NS	MSABC-NS
F1	6.52E−53	7.80E−57	**6.60E−63**	2.43E−58
F2	8.06E−50	3.88E−54	**1.52E−58**	5.20E−55
F3	4.59E−54	7.82E−58	**5.97E−64**	1.60E−58
F4	3.44E−58	6.40E−45	**2.36E−64**	3.01E−43
F5	9.32E−29	2.12E−32	7.22E−33	**1.96E−33**
F6	1.01E+00	4.55E+00	**2.96E−22**	1.84E−10
F7	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	4.10E−06	4.13E−06	**0.00E+00**	**0.00E+00**
F9	2.36E−02	2.61E−02	**3.45E−04**	6.06E−04
F10	**1.05E+00**	6.16E−01	2.81E+01	2.60E+01
F11	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F13	**0.00E+00**	**4.88E−17**	**0.00E+00**	**0.00E+00**
F14	**3.82E−04**	**3.82E−04**	**3.82E−04**	**3.82E−04**
F15	3.48E−14	3.50E−14	2.15E−15	**2.01E−15**
F16	**1.57E−32**	**1.57E−32**	**1.57E−32**	**1.57E−32**
F17	**1.35E−32**	**1.35E−32**	2.78E−24	**1.35E−32**
F18	3.86E−08	2.65E−09	9.91E−09	**4.27E−13**
F19	**1.35E−31**	**1.35E−31**	4.39E−03	**1.35E−31**
F20	2.23E−03	1.65E−04	6.56E−04	**1.53E−04**
F21	**−7.83E+01**	**−7.83E+01**	**−7.83E+01**	**−7.83E+01**
F22	**−2.86E+01**	**−2.86E+01**	**−2.86E+01**	**−2.86E+01**

4.3 Compared with Other State-of-the-Art ABCs

In order to further verify the performance of MSABC-NS, we compare it with other three state-of-the-art ABC variants, i.e., GABC, MABC and MPGABC. The brief introductions of these three ABC variants have been given in the Sect. 1. The results are shown in the Table 3 and the best results have been marked in **boldface**. As seen, MSABC-NS also achieved the best performance among the four involved algorithms.

Table 3. Experimental results of MABC, qABC, MPGABC and MSABC-NS.

Function	MABC	qABC	MPGABC	MSABC-NS
F1	4.82E−39	1.34E−15	6.52E−53	**2.43E−58**
F2	4.36E−36	2.70E−20	8.06E−50	**5.20E−55**
F3	3.20E−40	4.65E−22	4.59E−54	**1.60E−58**
F4	2.11E−35	7.13E−51	**3.44E−58**	3.01E−43
F5	5.01E−21	7.78E−23	9.32E−29	**1.96E−33**
F6	9.37E+00	2.89E+00	1.01E+00	**1.84E−10**
F7	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	6.71E−06	6.78E−06	4.10E−06	**0.00E+00**
F9	2.87E−02	2.91E−02	2.36E−02	**6.06E−04**
F10	3.42E+00	**7.15E−02**	1.05E+00	2.60E+01
F11	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F13	1.13E−15	**0.00E+00**	**0.00E+00**	**0.00E+00**
F14	**3.82E−04**	**3.82E−04**	**3.82E−04**	**3.82E−04**
F15	3.06E−14	3.10E−14	3.48E−14	**2.01E−15**
F16	**1.57E−32**	6.85E−13	**1.57E−32**	**1.57E−32**
F17	**1.35E−32**	2.00E−17	**1.35E−32**	**1.35E−32**
F18	**4.44E−17**	1.19E−16	3.86E−08	4.27E−13
F19	**1.35E−31**	2.60E−13	**1.35E−31**	**1.35E−31**
F20	**0.00E+00**	**0.00E+00**	2.23E−03	1.53E−04
F21	**−7.83E+01**	**−7.83E+01**	**−7.83E+01**	**−7.83E+01**
F22	**−2.86E+01**	**−2.86E+01**	**−2.86E+01**	**−2.86E+01**

5 Conclusion

In order to enhance the performance of the basic ABC algorithm, we proposed a multi-strategy ABC with the neighborhood search operator. In the multi-strategy mechanism, two different solution search equations are used through a simple IF-ELSE structure, which aims to take the advantages of different solution search equation. Furthermore, a global neighborhood search operator is introduced into our approach to improve the performance, and this operator is helpful to speed up the convergence rate while without losing diversity. Based on the 22 wildly used benchmark functions, the experimental results have shown the effectiveness of our approach.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Nos. 61603163 and 61876074) and the Science and Technology Foundation of Jiangxi Province (No. 20151BAB217007).

References

1. Tang, K.S., Man, K.F., Kwong, S., He, Q.: Genetic algorithms and their applications. IEEE Signal Process. Mag. **13**(6), 22–37 (1996)
2. Hunter, A., Chiu, K.S.: Genetic algorithm design of neural network and fuzzy logic controllers. Soft. Comput. **4**(3), 186–192 (2000)
3. Kennedy, J., Eberhart R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE (1995)
4. Kuo, R.J., Wang, M.H., Huang, T.W.: An application of particle swarm optimization algorithm to clustering analysis. Soft. Comput. **15**(3), 533–542 (2011)
5. Price, K., Storn, R., Lampinen, J.: Differential evolution: a practical approach to global optimization. In: ACM Computing Classification. Springer, Berlin (2005)
6. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report-TR06. Erciyes University (2005)
7. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. **39**(3), 459–471 (2007)
8. Zhu, G., Kwong, S.: Gbest-guided artificial bee colony algorithm for numerical function optimization. Appl. Math. Comput. **217**(7), 3166–3173 (2010)
9. Gao, W., Liu, S.: A modified artificial bee colony algorithm. Comput. Oper. Res. **39**(3), 687–697 (2012)
10. Karaboga, D., Gorkemli, B.: A quick artificial bee colony (qABC) algorithm and its performance on optimization problems. Appl. Soft Comput. **23**, 227–238 (2014)
11. Cui, L., et al.: Modified Gbest-guided artificial bee colony algorithm with new probability model. Soft. Comput. **22**, 2217–2243 (2018)
12. Wang, H., Wu, Z., Rahnamayan, S., Sun, H., Liu, Y., Pan, J.: Multi-strategy ensemble artificial bee colony algorithm. Inf. Sci. **279**, 587–603 (2014)
13. Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for differential evolution. In: IEEE Congress on Evolutionary Computation (2013)
14. Wang, H., Sun, H., Li, C., Rahnamayan, S., Pan, J.S.: Diversity enhanced particle swarm optimization with neighborhood search. Inform. Sci. **223**, 119–135 (2013)