Teaching-Learning-Based Artificial Bee Colony

Xu Chen[1(✉)] and Bin Xu[2]

[1] School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, Jiangsu, China
xuchen@ujs.edu.cn

[2] School of Mechanical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Abstract. This paper proposes a new hybrid metaheuristic algorithm called teaching-learning artificial bee colony (TLABC) for function optimization. TLABC combines the exploitation of teaching learning based optimization (TLBO) with the exploration of artificial bee colony (ABC) effectively, by employing three hybrid search phases, namely teaching-based employed bee phase, learning-based onlooker bee phase, and generalized oppositional scout bee phase. The performance of TLABC is evaluated on 30 complex benchmark functions from CEC2014, and experimental results show that TLABC exhibits better results compared with previous TLBO and ABC algorithms.

Keywords: Metaheuristic algorithm
Teaching learning based optimization · Artificial bee colony
Hybridization

1 Introduction

Metaheuristic search (MS) algorithms have received much attention regarding their potential as global optimization technique. Teaching learning based optimization (TLBO) [1] and artificial bee colony (ABC) [2] are two relative new MS algorithms. TLBO [3] is inspired by the teaching and learning process of a typical class, and it uses two operators namely teacher phase and a learner phase to search good solutions. ABC [2] is inspired by the intelligent foraging behavior of honey bees, and it uses three kinds of bees, namely employed bees, onlooker bees, and scout bees, to find good solutions. Both TLBO and ABC have aroused much interest in the last years and have been successfully applied to different kind of optimization problems [4–8].

Previous studies show that TLBO is good at exploitation [4], while ABC has a good exploration for global optimization [9]. However, a good search process needs to balance both exploration and exploitation; therefore, this paper proposes a hybrid teaching-learning-based artificial bee colony (TLABC) algorithm for global optimization. The proposed TLABC employs three hybrid search

Y. Tan et al. (Eds.): ICSI 2018, LNCS 10941, pp. 166–178, 2018.
https://doi.org/10.1007/978-3-319-93815-8_17

phases, namely teaching-based employed bee phase, learning-based onlooker bee phase, and generalized oppositional scout bee phase, which can efficiently combine the exploitation of TLBO with the exploration of ABC. TLABC is evaluated on 30 complex benchmark functions from CEC2014, and compared with eight well-established ABC and TLBO algorithms.

2 A Brief Introductions to TLBO and ABC

2.1 Teaching-Learning-Based Optimization

TLBO is a population-based optimization method which mimics the teaching and learning processes of a typical class [3]. The optimization process of TLBO is divided into two stages: teacher phase and learner phase.

In the teacher phase, the teacher provides knowledge to the learners to increase the mean result of the class. The learner with the best fitness in the current generation is identified as the teacher $\boldsymbol{x}_{teacher}$, and the mean position is represented as $\boldsymbol{x}_{mean}=\frac{1}{NP}\sum_{i=1}^{NP}\boldsymbol{x}_i$. The position of each learner is updated by Eq. (1):

$$\boldsymbol{x}_i^{new} = \boldsymbol{x}_i^{old} + \boldsymbol{rand} \cdot (\boldsymbol{x}_{teacher} - T_F \cdot \boldsymbol{x}_{mean}) \tag{1}$$

where $\boldsymbol{x}_i^{new}$ and $\boldsymbol{x}_i^{old}$ are the i-th learners new and old positions, respectively; $\boldsymbol{rand}$ is a random vector uniformly distributed within $[0,1]^D$; $T_F = round[1 + rand(0,1)]$ is the teacher factor; and its value is heuristically set to either 1 or 2. If $\boldsymbol{x}_i^{new}$ is better than $\boldsymbol{x}_i^{old}$, $\boldsymbol{x}_i^{new}$ is accepted and flowed to learner phase, otherwise $\boldsymbol{x}_i^{old}$ is unchanged.

In the learner phase, each earner randomly interacts with other different learners to further improve his/her performance. Learner $\boldsymbol{x}_i$ randomly selects another learner $\boldsymbol{x}_j (j \neq i)$ and the learning process can be expressed by Eq. (2):

$$\boldsymbol{x}_i^{new} = \begin{cases} \boldsymbol{x}_i^{old} + \boldsymbol{rand} \cdot (\boldsymbol{x}_i - \boldsymbol{x}_j), & \text{if } f(\boldsymbol{x}_i) \leq f(\boldsymbol{x}_j) \\ \boldsymbol{x}_i^{old} + \boldsymbol{rand} \cdot (\boldsymbol{x}_j - \boldsymbol{x}_i), & \text{if } f(\boldsymbol{x}_j) > f(\boldsymbol{x}_i) \end{cases} \tag{2}$$

2.2 Artificial Bee Colony

ABC is a swarm intelligence algorithm inspired on the foraging behavior of honey bee swarms [10]. It implements a cycle of the employed bee phase, onlooker bee phase and scout bee phase to search good solutions.

The ABC algorithm starts with randomly producing food sources:

$$x_{ij} = x_{\min,j} + (x_{\max,j} - x_{\min,j}) \cdot rand \tag{3}$$

where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$, $i \in \{1, 2, \cdots, NP\}$ represents the i-th solution; $\boldsymbol{x}_{\min,j}$ and $\boldsymbol{x}_{\min,j}$ are the lower and upper bounds for the dimension j, respectively; $rand$ is a random real number in the range of [0, 1]. The fitness values of the food sources are then calculated as:

$$fit(\boldsymbol{x}_i) = \begin{cases} \frac{1}{1+f(\boldsymbol{x}_i)} & \text{if } f(\boldsymbol{x}_i) \geq 0 \\ 1 + |f(\boldsymbol{x}_i)| & \text{otherwise} \end{cases} \tag{4}$$

where $f(\boldsymbol{x}_i)$ is the objective function value of $\boldsymbol{x}_i$.

In the employed bee phase, each employed bee performs a modification on the position of the food source by randomly selecting a neighboring food source. A new food source can be generated from the old food source $\boldsymbol{u}_i = (u_{i1}, u_{i2}, \cdots, u_{iD})$ as follows:

$$u_{ij} = x_{ij} + \psi \cdot (x_{ij} - x_{kj}) \tag{5}$$

where $k \in \{1, 2, \cdots, NP\}$ is randomly chosen index and must be different from i; ψ is a random number in the range $[-1, 1]$.

In the onlooker bee phase, an onlooker bee selects a food source to seek out ac-cording to the selection probability p, which is calculated as

$$p_i = \frac{fit(\boldsymbol{x}_i)}{\sum_{i=1}^{SN} fit(\boldsymbol{x}_i)} \tag{6}$$

where fit is the fitness value of the solution, which is calculated as using Eq. (4). After the onlooker bee selects a food source $\boldsymbol{x}_s$ to seek out, a candidate food source position $\boldsymbol{u}_s = (u_{s1}, u_{s2}, \cdots, u_{sD})$ will be produced by using Eq. (5).

If the food source position of the employed bees cannot be further improved through a given number of steps (*limit*) in the ABC algorithm, this employed bee becomes a scout bee. The new random food source position (scout bee) will be generated from the Eq. (3).

The candidate solution is compared with the old one. If the new food source has a better quality than the old source, then the old source is replaced by the new one. Otherwise, the old source is retained.

3 Proposed Teaching-Learning-Based Artificial Bee Colony

TLBO is good at exploitation, but its exploration is relative poor for complex problems [4]. On the other hand, ABC has a good exploration for global optimization but poor at exploitation [9]. In order to balance the exploration and the exploitation during the searching process, this section proposes a hybrid teaching-learning artificial bee colony (TLABC) algorithm, which effectively combines the exploitation of TLBO with the exploration of ABC.

TLABC starts with initializing NP food sources $\boldsymbol{x}_i = (x_{i1}, \cdots, x_{ij}, \cdots, x_{iD})$, and calculates the fitness values using Eq. (4). Then it uses three hybrid search phases to find good solutions: (1) teaching-based employed bee phase, (2) learning-based onlooker bee phase, and (3) generalized oppositional scout bee phase. The details of these three phases are described as follows.

3.1 Teaching-Based Employed Bee Phase

In the teaching-based employed bee phase, each employed bee searches a new food source $\boldsymbol{u}_i = (u_{i1}, u_{i2}, \cdots, u_{iD})$ using a hybrid TLBO teaching strategy:

$$u_{i,d} = \begin{cases} x_{i,d}^{old} + rand_2 \cdot (x_{teacher,d} - T_F \cdot x_{mean,d}) & \text{if } rand_1 < 0.5 \\ x_{r1,d} + F \cdot (x_{r2,d} - x_{r3,d}) & \text{otherwise} \end{cases} \quad (7)$$

where r_1, r_2, and $r_3 (r_1 \neq r_2 \neq r_3 \neq i)$ are integers randomly selected from $\{1, 2, \cdots, NP\}$; $d \in \{1, 2, \cdots, D\}$; $rand_1$ and $rand_2$ are two random numbers uniformly distributed within $[0, 1]$; and F is a scale factor in $[0, 1]$. If $\boldsymbol{u}_i$ is better than $\boldsymbol{x}_i$, then $\boldsymbol{u}_i$ is used to replace $\boldsymbol{x}_i$.

The hybrid TLBO teaching strategy using Eq. (7) can be viewed as a hybrid of basic teaching strategy of TLBO and mutation operator of differential evolution [11], and it is illustrated in Fig. 1.

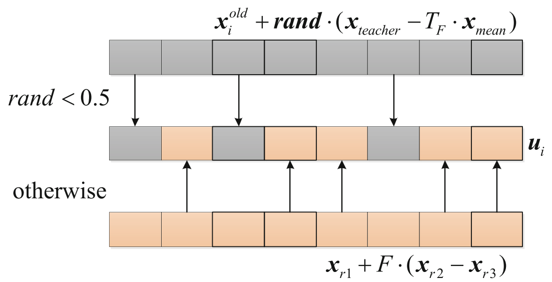

Fig. 1. Hybrid TLBO teaching strategy.

Remark 1: In the basic TLBO teaching strategy, all individuals use the same differential vector $(\boldsymbol{x}_{teacher} - T_F \cdot \boldsymbol{x}_{mean})$ to update the positions, so the diversity of the search directions is poor. By contrast, the hybrid TLBO teaching strategy uses a combination of TLBO teaching strategy and differential evolution mutation operator, which can improves the diversity of search directions greatly, and enhance the search ability of the proposed algorithm.

3.2 Learning-Based Onlooker Bee Phase

After the teaching-based employed bee phase, TLABC enters into the learning-based onlooker bee phase. In the learning-based onlooker bee phase, an onlooker bee selects a food source $\boldsymbol{x}_s$ to seek out according to the selection probability p, which is calculated using Eq. (6).

Then, the onlooker bee searches new food sources using the learning strategy of TLBO as follows:

$$\boldsymbol{u}_s = \begin{cases} \boldsymbol{x}_s + \boldsymbol{rand} \cdot (\boldsymbol{x}_s - \boldsymbol{x}_j), & \text{if } f(\boldsymbol{x}_s) \leq f(\boldsymbol{x}_j) \\ \boldsymbol{x}_s + \boldsymbol{rand} \cdot (\boldsymbol{x}_j - \boldsymbol{x}_s), & \text{if } f(\boldsymbol{x}_j) > f(\boldsymbol{x}_s) \end{cases} \quad (8)$$

where $j \in \{1, 2, \cdots, NP\}$ and $j \neq s$. If $\boldsymbol{u}_s$ is better than $\boldsymbol{x}_s$, then $\boldsymbol{u}_s$ is used to replace $\boldsymbol{x}_s$.

3.3 Generalized Oppositional Scout Bee Phase

After the learning-based onlooker bee phase, TLABC enters into the generalized oppositional scout bee phase. The generalized oppositional scout bee phase is proposed in [12], and it uses generalized opposition-based learning strategy [13] to enhance the basic scout bee phase. In this phase, if a food source $\boldsymbol{x}_i$ cannot be improved further for at least *limit* times, it is considered to be exhausted and would be abandoned. Then, a new random candidate solution $\boldsymbol{x}_i = (x_{i1}, \cdots, x_{ij}, \cdots, x_{iD})$ together with its generalized oppositional solution $\boldsymbol{x}_i^{GO} = (x_{i1}^{GO}, x_{i2}^{GO}, \cdots, x_{iD}^{GO})$ are generated using Eqs. (3) and (9) respectively.

$$x_{ij}^{GO} = k \cdot (a_j + b_j) - x_{ij} \tag{9}$$

where k is a random number in $[0, 1]$, and $a_j = \max_i(x_{ij}), \quad b_j = \min_i(x_{ij})$.

The better solution between $\boldsymbol{x}_i$ and $\boldsymbol{x}_i^{GO}$ are used to replace the old exhausted food source:

$$\boldsymbol{x}_i = \begin{cases} \boldsymbol{x}_i, & \text{if } f(\boldsymbol{x}_i) \leq f(\boldsymbol{x}_i^{GO}) \\ \boldsymbol{x}_i^{GO}, & \text{if } f(\boldsymbol{x}_i) > f(\boldsymbol{x}_i^{GO}) \end{cases} \tag{10}$$

3.4 Framework of TLABC

Based on the three search phases described above, the framework of TLABC can be summarized in Fig. 2.

Remark 2: TLABC employs the three bee phases of ABC to search solutions, which is beneficial for global exploration. Meanwhile, TLABC uses the improved TLBO search equations and opposition-based learning strategy in the three bee phases, which is benefit for local exploitation. Therefore, it is hopeful that TLABC can provide a better balance between exploration and exploitation compared with the TLBO and ABC algorithms.

4 Experimental Results and Analysis

In this section, the proposed TLABC is evaluated on 30 benchmark functions from CEC2014 competition [14] and compared with well-established TLBO and ABC algorithms. The benchmark functions can be categorized into four groups: (1) G1: unimodal functions (F01–F03); (2) G2: simple multimodal functions (F04–F16); (3) G3: hybrid functions (F17–F22); and (4) G4: composition functions (F23–F30). More details for these functions can be found in [14]. The evaluation is performed under 30 variables, i.e., $D = 30$. The maximum number of functional evaluations $MaxFES = 10^4 \times D$ was used to terminate TLABC and the compared algorithms. All algorithms are run 30 times independently.

Algorithm 1. Teaching-learning-based artificial bee colony

Stage 1: Initialization
  For each index $i = 1, 2, \cdots, NP$
    Initialize the position of individual $\boldsymbol{x}_i$ using Eq. (3);;
    Calculate the objective function value $f(\boldsymbol{x}_i)$;
    Calculate the fitness $fit(\boldsymbol{x}_i)$ using Eq. (4);
    Set $trial(i) = 0$ for each individual;
  End For
Stage 2: Teaching-based employed bee phase
  For each index $i = 1, 2, \cdots, NP$
    Generate a new candidate solution $\boldsymbol{u}_i$ by hybrid TLBO teaching strategy using Eq.(7);
    Calculate the objective function value $f(\boldsymbol{u}_i)$;
    Calculate the fitness value $fit(\boldsymbol{u}_i)$ using Eq. (4);
    Apply a greedy selection process between $\boldsymbol{u}_i$ and $\boldsymbol{x}_i$ to select the better one;
    If solution $\boldsymbol{x}_i$ does not improve, $trial(i) = trial(i) + 1$, otherwise $trial(i) = 0$;
  End For
Stage 3: Learning-based on looker bee phase
    Calculate the selection probability p_i using Eq.(6);
    For each index $i = 1, 2, \cdots, NP$
      Select a solution $\boldsymbol{x}_s$ using the roulette method according to the probability p_i ;
      Generate a new candidate solution $\boldsymbol{u}_s$ by TLBO learning strategy using Eq.(8) ;
      Calculate the objective function value $f(\boldsymbol{u}_s)$
      Calculate the fitness value $fit(\boldsymbol{u}_s)$ using Eq. (4);
      Apply a greedy selection process between $\boldsymbol{u}_s$ and $\boldsymbol{x}_s$ to select the better one.
      If solution $\boldsymbol{x}_s$ does not improve, $trial(s) = trial(s) + 1$, otherwise $trial(s) = 0$.
    End For
Stage 4: Generalized oppositional scout bee phase
  If $\max(trial(i)) \geq \text{limit}$, **Then**
      Generate a new random candidate solution $\boldsymbol{x}_i$ and its generalized oppositional solution $\boldsymbol{x}_i^{GO}$;
      Replace the old $\boldsymbol{x}_i$ with the better one between the new $\boldsymbol{x}_i$ and $\boldsymbol{x}_i^{GO}$;
  End If
Stage 5: If the stop criterion is satisfied, stop and output the best solution achieved so far. Otherwise, return to Step 2.

Fig. 2. Framework of TLABC.

4.1 Compared with TLBO Algorithms

TLABC is first compared with four well-known TLBO algorithms, they are basic TLBO [3], nonlinear inertia weighted TLBO (NIWTLBO) [15], TLBO with learning experience of other learners (LETLBO) [16], and generalized oppositional TLBO (GOTLBO) [5]. The parameters settings for TLABC and the compared TLBO algorithms are listed in Table 1.

Table 1. Parameter settings for TLABC and the compared TLBO algorithms

Algorithm	Parameter settings
TLBO	Population size $NP = 50$
NIWTLBO	$NP = 50$, inertia weight $w = 0 \sim 1.0$
LETLBO	$NP = 50$
GOTLBO	$NP = 50$, jumping rate $Jr = 0.3$
TLABC	$NP = 50$, $limit = 200$, scale factor $F = rand(0, 1)$

Table 2 compares TLABC and the TLBO algorithms on four groups of benchmark functions. Following [17], each cell of the win-draw-loss table (see Table 2) consists of three numbers in $\alpha - \beta - \gamma$ style. In each triplet, α denotes the number of functions on the corresponding group which the TLABC performs statistically better than its competitor. The next number β shows how many times TLABC performs statistically similar its competitor. And, γ denotes the number of functions that TLABC performs statistically worse than its competitor. Note that in this table two algorithms are considered to be significantly different if the p-value of Wilcoxon rank-sum test is less than 0.05, and statistically similar otherwise.

Table 2. The win-draw-loss statistics results between TLABC and the compared TLBO algorithms

	TLBO	NIWTLBO	LETLBO	GOTLBO
G1	3-0-0	3-0-0	3-0-0	2-1-0
G2	9-2-2	9-2-2	7-3-3	8-3-2
G3	3-2-1	3-0-3	3-1-2	2-3-1
G4	5-2-1	3-2-3	6-2-0	1-2-5
Total	20-6-4	18-4-8	19-6-5	13-9-8

First, for unimodal functions F01–F03, TLABC performs significantly better than TLBO, NIWTLBO, and LETLBO on all 3 functions, and better than GOTLBO on 2 functions.

Second, for simple multimodal functions F04–F16, TLABC is significantly better than TLBO, NIWTLBO, and LETLBO on 9, 9, 7, and 8 functions, respectively. It is significantly worse than TLBO, NIWTLBO, and LETLBO on 2, 2, 3, and 2 functions, and similar to them on 2, 2, 3, and 3 functions, respectively.

Thirdly, for hybrid functions F17–F22, the performance of TLABC is better than TLBO, LETLBO and GOTLBO, while similar to NIWTLBO.

Finally, with regard to composition functions F23–F30, TLABC performs better than TLBO and LETLBO, similar to NIWTLBO, while worse than GOTLBO.

In summary, compared with the TLBO algorithms, TLABC shows the best overall performance on the unimodal functions, simple multimodal functions and hybrid functions. On composition functions, GOTLBO shows the best performance, TLABC and NIWTLBO the second. Based on the comparisons, it can be seen clearly that TLABC significantly improves the performance of TLBO on multimodal functions (i.e. the functions in Group 2). It should be attributed to the utilization of the three bee phases in TLABC, which is beneficial for the global exploration of TLABC. Meanwhile, the performance of TLABC is also very competitive on the other three group functions compared with TLBO algorithms.

4.2 Compared with ABC Algorithms

TLABC is also compared with four well-established ABC algorithms, they are basic ABC [2], gbest-guided ABC (GABC) [9], modified ABC(MABC) [18], and Gaussian bare-bones ABC (GBABC) [12]. The parameters settings for the TLABC and compared ABC algorithms are listed in Table 3.

Table 4 shows the win-draw-loss statistics results between TLABC and the compared ABC algorithms.

Table 3. Parameter settings for TLABC and the compared TLBO algorithms

Algorithm	Parameter settings
ABC	Population size $NP = 100$, $limit = 100$
GABC	$NP = 100$, $limit = 100$, $C = 1.5$
MABC	$NP = 100$, $limit = 100$, modification rate $MR = 0.4$, scaling factor $SF = 1$
GBABC	$NP = 100$, $limit = 100$, crossover rate $CR = 0.3$

Table 4. The win-draw-loss statistics results between TLABC and the compared ABC algorithms

	ABC	GABC	MABC	GBABC
G1	3-0-0	3-0-0	2-1-0	3-0-0
G2	3-2-8	3-2-8	5-2-6	3-2-8
G3	5-0-1	5-0-1	3-0-3	3-3-0
G4	7-0-1	6-2-0	5-2-1	4-3-1
Total	20-6-4	18-4-8	19-6-5	13-9-8

First, for unimodal functions F01–F03, TLABC performs significantly better than ABC, GABC, and GBABC on all 3 functions. TLABC performs significantly better than MABC on 2 functions, and similarly on 1 function.

Second, for simple multimodal functions F04–F16, the performance of TLABC is not very satisfactory. It performs worse than the other ABC.

Thirdly, for hybrid functions F17–F22, TLABC performs significant better than ABC, GABC, MABC, and GBABC on 5, 5, 3, and 3 functions, respectively. It is performs worse than ABC, GABC, MABC and GBABC on 1, 1, 3, and 0 functions, and similar to them on 0, 0, 0, and 3 functions, respectively.

Finally, with regard to composition functions F23–F30, TLABC shows the best performance on most of functions. It also outperforms the ABC algorithms on most of functions.

In summary, TLABC shows the best overall performance on the unimodal functions, hybrid functions, and composition functions. For the simple multimodal functions, the performance of TLABC is not very satisfactory, but it also gets the best results on some multimodal functions.

From the comparisons, we can see that TLABC significantly improves the performance of ABC on unimodal functions (i.e. the functions in Group 1). This can be attributed to the use of the improved TLBO search equations and opposition-based learning strategy in TLABC, which enhances the global exploitation of TLABC. However, it sacrifices the exploration of the algorithm, as the performance of TLABC is not very satisfactory on simple multimodal functions compared with the ABC algorithms. Fortunately, TLABC achieves a very excellent performance on the hybrid functions and composition functions; therefore, the introduction of TLBO search equations and opposition-based learning strategy is useful for the overall performance enhancement of ABC.

4.3 Multiple-Problem Statistical Comparison

Table 5 shows the results of multiple-problem Wilcoxon test [19] between TLABC and the compared TLBO and ABC algorithms. It can be seen from the Table 5 that TLABC attains higher positive-ranks ($R+$) than negative-ranks ($R-$) compared with all the ABC and TLBO algorithms. This means that TLABC is overall better than the compared algorithms for all functions. Also, there are significant differences among TLABC, ABC, TLBO, NIWTLBO, LETLBO and GOTLBO when $\alpha = 0.05$ and $\alpha = 0.1$.

Table 5. Results of Multiple-problem Wilcoxon test between TLABC and the compared TLBO and ABC algorithms

	$R+$	$R-$	p-value	$\alpha = 0.05$	$\alpha = 0.1$
TLABC vs TLBO	375	60	3.32E-04	Yes	Yes
TLABC vs NIWTLBO	318	117	2.91E-02	Yes	Yes
TLABC vs LETLBO	331	104	1.29E-02	Yes	Yes
TLABC vs GOTLBO	328	107	1.57E-02	Yes	Yes
TLABC vs ABC	331	134	4.27E-02	Yes	Yes
TLABC vs GABC	311	154	1.09E-01	No	No
TLABC vs MABC	284	181	0.2	No	No
TLABC vs GBABC	256	209	0.2	No	No

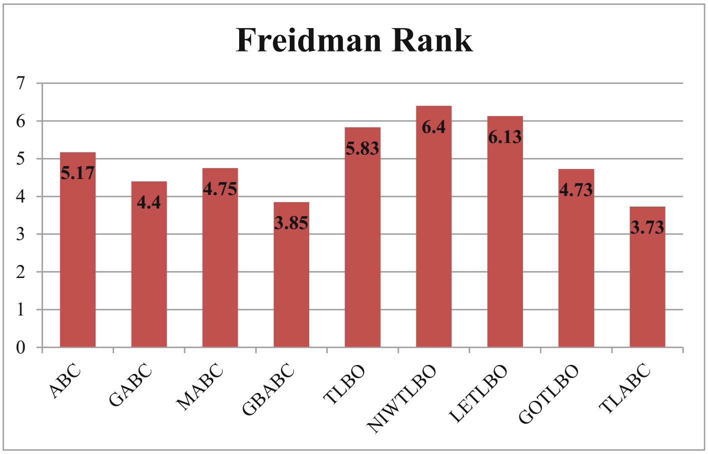

Fig. 3. Friedman rank of TLABC and the compared algorithms.

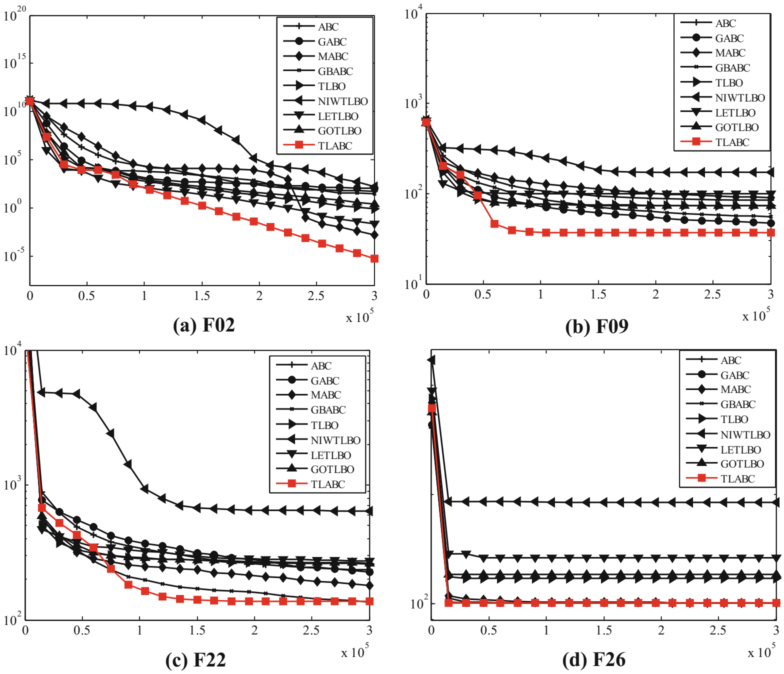

Fig. 4. Convergence graphs of TLABC and the compared algorithms on four typical functions.

Figure 3 shows the results of the Friedman rank test among TLABC and the compared algorithms. As shown in Fig. 3, TLABC gets the best rank, followed in order by GBABC, GABC, GOTLBO, MABC, ABC, TLBO, LETLBO, and NIWTLBO.

Figure 4 plots the Convergence graphs of TLABC and the compared algorithms on four typical functions F02, F09, F22, and F26. Overall, TLABC converge faster than the compared ABC and TLBO algorithms on these four functions.

5 Conclusion

Teaching learning based optimization (TLBO) and artificial bee colony (ABC) are two metaheuristic algorithms which have aroused great interests in recent years and have demonstrated their effectiveness on a wide variety of optimization problems. TLBO employs teaching and learning operators to search solutions, and its production operators are good at exploitation. ABC uses three search phases, namely employed bee phase, onlooker bee phase, and scout bee phase, to explore solutions. ABC is good at exploration; however, its exploitation ability is relative poor. In this paper, we have proposed a new hybrid algorithm named teaching-learning-based artificial bee colony (TLABC), which employs the teaching and learning operators of TLBO and the three bee search phases of ABC. The proposed TLABC is evaluated on the CEC2014 benchmark functions, and compared with previous eight well-known TLBO and ABC algorithms. Based on the experimental results, we can conclude that:

(1) TLABC significantly improves the performance of TLBO algorithms on multimodal functions. It should be attributed to the utilization of the three bee phases in TLABC, which is beneficial for the global exploration of TLABC.
(2) TLABC significantly improves the performance of ABC on unimodal functions. This can be attributed to the use of the improved TLBO search equations and opposition-based learning strategy in TLABC, which enhances the global exploitation of TLABC.
(3) TLABC can provide a better balance between exploration and exploitation compared with the TLBO and ABC algorithms. The overall performance of TLABC is better than all the compared TLBO and ABC algorithms on 30 CEC2014 benchmark functions, which demonstrates the superiority of the proposed hybrid algorithm.

Acknowledgements. This work was supported in part by Natural Science Foundation of Jiangsu Province (Grant No. BK 20160540), China Postdoctoral Science Foundation (Grant No. 2016M591783), and National Natural Science Foundation of China (Grant No. 61703268).

References

1. Rao, R.V., Savsani, V.J., Vakharia, D.: Teachinglearning-based optimization: a novel method for constrained mechanical design optimization problems. Comput. Aided Des. **43**, 303–315 (2011)
2. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global Optim. **39**, 459–471 (2007)
3. Rao, R., Savsani, V., Vakharia, D.: Teachinglearning-based optimization: an optimization method for continuous non-linear large scale problems. Inf. Sci. **183**, 1–15 (2012)
4. Zou, F., Wang, L., Hei, X., Chen, D., Yang, D.: Teachinglearning-based optimization with dynamic group strategy for global optimization. Inf. Sci. **273**, 112–131 (2014)
5. Chen, X., Yu, K., Du, W., Zhao, W., Liu, G.: Parameters identification of solar cell models using generalized oppositional teaching learning based optimization. Energy **99**, 170–180 (2016)
6. Yu, K., Wang, X., Wang, Z.: Constrained optimization based on improved teaching-learning-based optimization algorithm. Inf. Sci. **352**, 61–78 (2016)
7. Oliva, D., Cuevas, E., Pajares, G.: Parameter identification of solar cells using artificial bee colony optimization. Energy **72**, 93–102 (2014)
8. Xiang, Y., Peng, Y., Zhong, Y., Chen, Z., Lu, X., Zhong, X.: A particle swarm inspired multi-elitist artificial bee colony algorithm for real-parameter optimization. Comput. Optim. Appl. **57**, 493–516 (2014)
9. Zhu, G., Kwong, S.: Gbest-guided artificial bee colony algorithm for numerical function optimization. Appl. Math. Comput. **217**, 3166–3173 (2010)
10. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artif. Intell. Rev. **42**, 21–57 (2014)
11. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. **15**, 4–31 (2011)
12. Zhou, X., Wu, Z., Wang, H., Rahnamayan, S.: Gaussian bare-bones artificial bee colony algorithm. Soft. Comput. **20**(3), 907–924 (2016)
13. Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., Ventresca, M.: Enhancing particle swarm optimization using generalized opposition-based learning. Inf. Sci. **181**, 4699–4714 (2011)
14. Liang, J., Qu, B., Suganthan, P.: Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore, Computational Intelligence Laboratory (2013)
15. Wu, Z.-S., Fu, W.-P., Xue, R.: Nonlinear inertia weighted teaching-learning-based optimization for solving global optimization problem. Comput. Intell. Neurosci., 87 (2015)
16. Zou, F., Wang, L., Hei, X., Chen, D.: Teachinglearning-based optimization with learning experience of other learners and its application. Appl. Soft Comput. **37**, 725–736 (2015)
17. Kazimipour, B., Omidvar, M.N., Li, X., Qin, A.K.: A novel hybridization of opposition-based learning and cooperative co-evolutionary for large-scale optimization. In: IEEE Congress on Evolutionary Computation (CEC), pp. 2833–2840. IEEE (2014)

18. Akay, B., Karaboga, D.: A modified artificial bee colony algorithm for real-parameter optimization. Inf. Sci. **192**, 120–142 (2012)
19. Garca, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the CEC2005 special session on real parameter optimization. J. Heuristics **15**, 617–644 (2009)