An Experience Information Teaching–Learning-Based Optimization for Global Optimization

Zhuo Wang, *Member, IEEE*, Renquan Lu, *Member, IEEE*, Debao Chen, and Feng Zou

***Abstract*—Teaching–learning-based optimization (TLBO) is an intelligent optimization algorithm with relatively fewer parameters that should be determined in updating equations. For solving complex optimization problems, the local optima often appear in the evolution. To decrease the possibility of this phenomenon, a novel TLBO variant (EI-TLBO) with experience information (EI) and differential mutation is presented. In the method, neighborhood information (the best individual NTeacher and the mean individual NMean) of each learner's neighbors is introduced to improve the exploration capability. The EI before the current iteration of each learner is introduced to make him or her accurately judge the learning behavior in future. In addition, instead of duplicate elimination to maintain the diversity of population at the end of each generation in the original TLBO, differential mutation is introduced to maintain the diversity of learners during the iterative learning process. The main contribution of this paper is to improve the convergence speed and accuracy by introducing neighborhood topology structure, EI, and differential mutation. The efficiency of the proposed algorithm is evaluated on 46 benchmark functions, among which 27 functions are selected from CEC2013. Its performance is compared with those of six other reported EAs. The results indicate that EI-TLBO algorithm can achieve superior performance.**

***Index Terms*—Differential mutation, experience information (EI), global optimization, teaching–learning-based optimization (TLBO).**

I. Introduction

FUZZY technology [1], [2], neural networks [3]–[6], support vector machines [7], etc. have been used to deal with complex nonlinear problems, and some successful results are derived. With the rapid development of technology and science, some real-world engineering problems can be converted as global optimization problems. To solve these problems, many nature-inspired optimization algorithms have attracted growing research interest from many research fields and been widely developed over the years. These population-based optimization algorithms are mainly divided into two classes: 1) evolutionary algorithms (EAs) and 2) swarm intelligence (SI). EAs [8]–[12] have originated from the natural evolution phenomena and principles, and SIs are inspired from the social character and behavior [13]–[16] of living things. These population-based optimization algorithms have also been successfully dealt with in some kinds of real-world optimization problems for decades.

TLBO is one of the SI algorithms whose framework is simple and the parameters that should be determined in updating equations are relatively fewer. Because of these characteristics, the TLBO algorithm has attracted more and more attention, and large amounts of existing TLBO variant algorithms make TLBO feasible and promising for complex optimization problems. However, the learners of the original TLBO algorithm acquire knowledge from the teacher of the whole class in the teacher phase and learn from other learners randomly chosen from the class in the learner phase. Learners do not make use of experience information (EI) obtained in the iterative learning process. In addition, the diversity of learners will be degraded with the increasing iteration of evolution. The diversity of population is maintained by means of the duplicate elimination in the original TLBO algorithm. However, this method is not differential to restart if the number of duplicate eliminations is large, and then it might decrease the convergence speed of the TLBO algorithm.

At the same time, the existing knowledge provides us valuable information on the behaviors of the individuals.

1) From the literature of social psychology [17], each individual belongs to a certain social neighborhood, and those individuals tend to imitate their behavior on other individuals' behavior from the same neighborhood. It also indicates that each individual is influenced by his/her neighbors. From the point of searching the solution space, it helps one to find better solutions by means of searching the neighborhoods of individuals. Hence, neighborhood topology technique is often introduced into the population-based stochastic optimization algorithms to ameliorate the exploration ability of individuals.

Manuscript received June 8, 2015; revised July 12, 2015; accepted July 18, 2015. Date of publication January 22, 2016; date of current version August 16, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61572224 and Grant 61304082, in part by the National Science Fund for Distinguished Young Scholars under Grant 61425009, in part by the Major Project of Natural Science Research in Anhui Province under Grant KJ2015ZD36, and in part by the Natural Science Foundation of Anhui Province under Grant 1308085MF82. This paper was recommended by Associate Editor Z. Li. *(Corresponding author: Renquan Lu.)*

Z. Wang is with the Fok Ying Tung Graduate School, Hong Kong University of Science and Technology, Hong Kong (e-mail: zhuowang@ust.hk).

R. Lu is with the Institute of Information and Automation, Hangzhou Dianzi University, Hangzhou 310000, China (e-mail: rqlu@hdu.edu.cn).

D. Chen and F. Zou are with the School of Physics and Electronic Information, Huaibei Normal University, Huaibei 235000, China (e-mail: chendb_8@163.com; zfemail@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2015.2503406

2) In the real world, the individuals often record their historical experience, and these experiences can help them in accurately judging on the behavior in future. Inspired by this phenomenon, some researchers introduced these mechanisms into the SI algorithm [18], [19], which can improve the performance of the algorithm. In fact, each learner is always learning based on his previous knowledge. Hence, EI so far can be introduced into TLBO to improve the learning efficiency of learners.
3) Individuals become more similar through mutual interaction among themselves, as they influence and imitate each other. From a biological point of view [20], societies with diverse individuals can perform more complex tasks. Diversity is a significant feature of the social complexity that results in diverse and differentiated individuals. The more diverse the society, the easier is its dealing with complex tasks. Because of its advantage of being unbiased to any prior given guider, differential evolution (DE) can maintain the diversity of individuals [21]. Hence, differential mutation strategy can be utilized to maintain the diversity of individuals because of not being biased toward any prior defined guider.

Motivated by these considerations, in this paper, an improved TLBO algorithm with EI and differential mutation (EI-TLBO) is presented to improve the global performance of TLBO. The main contribution of this paper is shown in three aspects.

1) The EI is introduced into TLBO to help learners judge the learning behavior in future. In this rule, EI obtained before the current iterative learning process, which is the same as the inertia component in the canonical particle swarm optimization (PSO), can be considered as the learning direction and tendency information of a learner.
2) The ring-neighborhood topology is adopted in the EI-TLBO algorithm and the neighborhood information of the learner is used to improve their exploration capability.
3) Instead of duplicate elimination at the end of each iteration, differential mutation, which is mainly based on the distance and direction information, is introduced into EI-TLBO to ameliorate the diversity of the swarm during the whole evolution.

The remaining parts of the paper are described as follows. In Section II, we provide a brief introduction of the original TLBO algorithm. Some variants of TLBO are briefly reviewed in Section III. The designing procedure of EI-TLBO is described in Section IV. Section V presents the experiments and testing results of different algorithms along with the statistical tests. Some conclusions and future works are given in Section VI.

II. Standard Teaching–Learning-Based Optimization Algorithm

The solutions of TLBO [16] are represented by a group of learners, and the optimization process TLBO mimics the knowledge-disseminating method in a classroom. In this method, all learners construct a swarm and the best learner of the current population is considered as the teacher. All learners perform two learning-evolution processes: 1) teacher phase and 2) learner phase. The two major phases of standard TLBO are formulated as follows.

A. Teacher Phase

In the original TLBO, teacher phase is used to help the learners acquire knowledge from the current teacher and the mean solution of the class. In this phase, the teacher disseminates his or her knowledge to all learners of the class so as to improve the average grades of the whole class. The learners acquire knowledge from the class by using the difference between the teacher and the mean result of the current learners.

Assume that the ith learner is learner X_i, the best learner in the current population of learners is teacher, and the mean position of the class is expressed with mean. The updating process of each learner can be given as follows.

The difference between teacher and mean is given by [9]

$$\text{Difference\_Mean} = r \ * \ (\text{Teacher} - \text{TF} * \text{Mean}). \tag{1}$$

Then, the updating equation of the ith learner X_i in teacher phase can be given as follows [16]:

$$\text{new}X_i = X_i + \text{Difference\_Mean} \tag{2}$$

where TF is a teaching factor and can be either 1 or 2 randomly and r is a random number in the domain [0, 1].

B. Leaner Phase

In the original TLBO, the second part of the algorithm is the learner phase. During this phase, all learners enhance their performance by mutually interacting among themselves. The working of this phase can be done in a class by group discussions, presentations, formal communications, etc., and the goal is to provide a fitter communication chance for all the learners to randomly interact with their peers.

Assume that the ith learner is X_i, and the learner interacting with him/her is X_j, then the updating equation of the ith learner X_i in learner phase can be described as follows [16]:

$$\text{new}X_i = \begin{cases} X_i + r * (X_i - X_j) & \text{if } f(X_i) \text{ is better than } f(X_j) \\ X_i + r * (X_j - X_i) & \text{otherwise} \end{cases} \tag{3}$$

where r is a random number taken from [0, 1], $f(X_i)$ and $f(X_j)$ are the fitness values of the learners X_i and X_j, respectively.

III. Brief Review of TLBO

Among these population-based optimization algorithms, owing to its characteristics like simple concept, no algorithm-specific parameters, rapid convergence, easy implementation, TLBO has been widely extended to many research areas and some good performance has been achieved for various function optimizations and engineering problems. Some variants of TLBO were proposed in recent years.

TLBO is proposed by Rao *et al.* [16], [22] to deal with the optimization of mechanical design problems, and then it is utilized to solve large-scale nonlinear optimization problems. Rao and Patel [23] introduced elitism concept into the TLBO algorithm to improve the global optimization performance, and they also investigated the effect of elitism size on the algorithm. To deal with the optimization of planar steel frames, Amiri [24] presented a design of discrete TLBO. Toğan [25] used TLBO to solve the clustering problem and the clustering ability is a test on some well-known datasets. Niknam *et al.* [26] presented a θ-multiobjective TLBO algorithm to solve the dispatch problem in dynamic economic emission where the parameter of the system is designed by use of the phase angles. Rao and Patel [27] presented an improved TLBO with multiteachers and adaptive teaching factor for heat exchangers, and they also proposed an improved TLBO with tutorial training and self-motivated learning [28]. Degertekin and Hayalioglu [29] used TLBO for the optimization of four truss structures. Satapathy *et al.* [30] proposed a weighted TLBO to increase convergence rate of the algorithm. Based on the learning characters during the teaching–learning process in the real class, Zou *et al.* [31] introduced dynamic group strategy into TLBO to improve the algorithm's global optimization performance. Mandal and Roy [32] introduced quasi-oppositional-based learning into the original TLBO algorithm to improve its global performance, and then the power-flow problem with multiobjective is solved on the condition that the system has some constraints. Considering the difference ability of each learner, Camp and Farshchin [33] introduced a fitness-based weighted mean in the teaching phase and a refined student-updating process into TLBO to solve the geometry space trusses design problem. In [34], an efficient TLBO algorithm with ring-neighborhood topology information is proposed, and the exploration abilities of individuals are ameliorated. By making use of the statistical information, Zou *et al.* [35] presented a bare bones TLBO algorithm with the Gaussian sampling. Ghasemi *et al.* [36] proposed a reliable and effective hybrid optimization algorithm based on modified TLBO and double DE to deal with the optimal reactive power dispatch (ORPD) problem. Cheng [37] used TLBO to screen primers conformed to primer constraints. In [38], a novel TLBO algorithm called Gaussian BBTLBO algorithm and its modified version MGBTLBO is proposed for the ORPD problem. For solving flexible job-shop scheduling, Xu *et al.* [39] incorporated bi-phase crossover scheme and special local search operators into the TLBO to balance the exploration and exploitation capabilities. Kadambur and Kotecha [40] proposed an elitist TLBO-based optimization strategy, which overcomes the limitations of the mixed integer linear programming formulation.

As mentioned earlier, most methods increase the diversity of TLBO with different strategies, and the historical information of the learners cannot be considered, which implies that full use of the EI of learners is not made to guide them toward the promising areas. To improve the diversity and direction, a new TLBO variant with EI and differential mutation is proposed in the following section.

IV. TLBO With Experience Information and Differential Mutation (EI-TLBO)

In this section, a detailed description of EI-TLBO will be provided.

A. Overall Framework of EI-TLBO

In the first EI-TLBO algorithm, *NP* learners, which form the class, are randomly initialized. Next, neighborhood topology technique is designed for EI-TLBO to improve the exploration capability of learners. Then, the updating rules of learners are presented by use of EI obtained before the current iteration in teacher phase and learner phase to improve the learning efficiency of learners. Moreover, differential mutation is introduced to maintain the diversity of learners during each iteration. The complete framework of the EI-TLBO algorithm is given as shown in Fig. 1. The detailed operators of EI-TLBO algorithm are described as follows.

B. Neighborhood Topology of EI-TLBO

As mentioned earlier, neighborhood information of individuals helps in finding better solutions by searching the neighborhood areas of these individuals. Based on this idea, by using neighborhood information, the exploration capability of population-based algorithms can be effectively enhanced and the individuals' diversity of population can be maintained [41]–[43]. Our previous work [34] indicates that the ring-neighborhood topology information can ameliorate the exploration ability of learners and improve the performance of the algorithm. Hence, instead of utilizing the global information, the ring-neighborhood topology of learners is also used in the EI-TLBO algorithm so that the learner can fully utilize the information of his corresponding neighbors to avoid over-congestion around the local optima.

C. Updating Rules of Learners

In the natural world, the individuals generally record their experience, and these experiences can help them in accurately judging on the behavior in future. We know that each learner is always learning based on his previous knowledge. Hence, in our proposed EI-TLBO algorithm, EI obtained before the current iterative learning process is introduced into the updating formula. Because of the greedy selection during the updating process of learners, EI obtained so far can be considered as the learning direction and tendency information of a learner. In fact, this component is the same as the inertia component in the canonical PSO. Making use of EI so far might improve the global searching ability of learners. On the other hand, learners may also learn from other random learners and the difference between them. This learning rule is in fact a differential mutation. Because of making full use of being unbiased toward any prior given guider, this mutation helps in maintaining the diversity of learners. The following sections describe the updating process of learners.

1) Learning in Teacher Phase: In the proposed method, the learners are probabilistically learning by means of the random TLBO learning strategy or the differential mutation learning strategy. First, a probability P_m is set for learners. Then, a random probability r is generated for each learner. Here, the value of r is between 0 and 1. If $r < P_m$, the random TLBO learning strategy is adopted by the learner; otherwise, the differential mutation learning strategy is adopted.

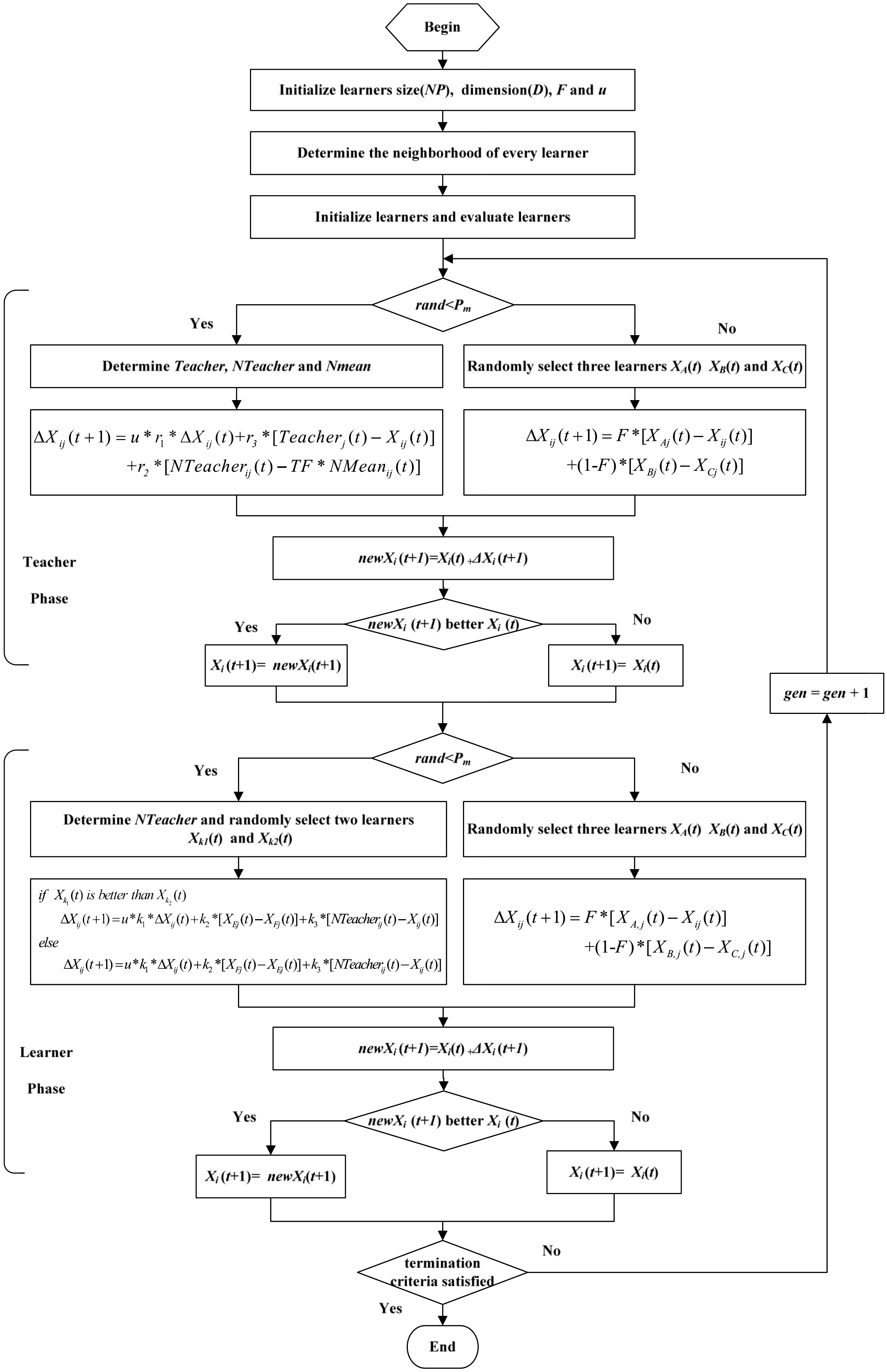


Fig. 1. Overall framework of EI-TLBO algorithm.

Algorithm 1 Teaching ( )

```
Begin % Teacher phase
for each learner X_i of the class
r=rand(.)
if r < p_m
Donate NTeacher_i and NMean_i from the neighborhood of each learner X_i;
TF = round(1+rand(0,1));
for j = 1 : D
Updating the learner according to Eq.4 and Eq.6
end for
else
for j = 1 : D
Updating the learner according to Eq.5 and Eq.6
end for
end if
Accept the new learner if it has better fitness value
end for
end
```

Algorithm 2 Learning ( )

```
Begin % Learner phase
for each learner X_i of the class
if r < p_m
Donate NTeacher_i from the neighborhood of each learner X_i;
TF = round(1+rand(0,1));
for j = 1 : D
Updating the learner according to Eq.7 and Eq.6
end for
else
for j = 1 : D
Updating the learner according to Eq.5 and Eq.6
end for
end if
Accept the new learner if it has better fitness value
end for
end
```

Assume that t is the iteration counter; Teacher is the current best learner in the whole class; NTeacher is the best individual from the learner's neighbors; NMean is the mean position of the learner's neighbors. Then, if $r < P_m$, the changing state of the ith learner can be described as follows:

$$\begin{aligned}\Delta X_{ij}(t+1) = u * k_1 * \Delta X_{ij}(t) + k_2 \\ * \left[\mathrm{NTeacher}_{ij}(t) - \mathrm{TF} * \mathrm{NMean}_{ij}(t)\right] \\ + k_3 * \left[\mathrm{Teacher}_j(t) - X_{ij}(t)\right]\end{aligned} \tag{4}$$

where u is the weighting factor, which affects the influence degree of the learner; k_1, k_2, and k_3 are random numbers between 0 and 1; TF is randomly set to either 1 or 2.

In the above updating rules, the first component $\Delta X_{ij}(t)$ contains EI of the ith learner. This component reflects the influence of the historical learning experience on the current learner. In the second component, instead of learning from the difference between Teacher and Mean of the whole class in TLBO, the learner acquires knowledge from NTeacher and NMean of his or her corresponding neighborhood. In addition, global learning is also added to the updating formula in the third part to help the learners acquire knowledge from the current teacher.

On the contrary, if $r \geq P_m$, the state changing of ith learner is shown as follows:

$$\begin{aligned}\Delta X_{ij}(t+1) = F * \left[X_{A,j}(t) - X_{ij}(t)\right] \\ + (1-F) * \left[X_{B,j}(t) - X_{C,j}(t)\right]\end{aligned} \tag{5}$$

where t is the iteration counter; A, B, C $(A \neq B \neq C \neq i)$ are random numbers in the range [1, NP]; NP is the size of the class; F is a mutation scaling factor and it affects the proportion of the two parts in (5).

According to (2), the updating equation of the ith learner in teacher phase of EI-TLBO is shown as follows:

$$\mathrm{new}X_{ij}(t+1) = X_{ij}(t) + \Delta X_{ij}(t+1). \tag{6}$$

As mentioned earlier, the pseudocode of teacher phase in the proposed EI-TLBO algorithm is given in Algorithm 1.

2) Learning in Learner Phase: Assume that t is the iteration counter; NTeacher is the teacher of the learner's corresponding neighborhood. Similar to the updating rules in teacher phase, if $r < P_m$, the state of the ith learner in the learning phase can be designed as follows:

$$\begin{aligned}&\text{if } X_E(t) \text{is better than } X_F(t)\\ &\quad \Delta X_{ij}(t+1) = u * k_1 * \Delta X_{ij}(t) + k_2 * \left[X_{Ej}(t) - X_{Fj}(t)\right] \\ &\qquad + k_3 * \left[\mathrm{NTeacher}_{ij}(t) - X_{ij}(t)\right]\\ &\text{else}\\ &\quad \Delta X_{ij}(t+1) = u * k_1 * \Delta\ X_{ij}(t) + k_2 * \left[X_{Fj}(t) - X_{Ej}(t)\right] \\ &\qquad + k_3 * \left[\mathrm{NTeacher}_{ij}(t) - X_{ij}(t)\right]\\ &\text{end}\end{aligned} \tag{7}$$

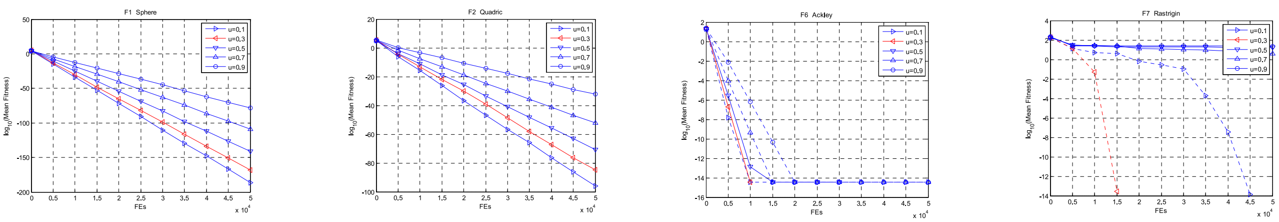
where u is the weighting factor, which affects the influence degree of the learner with EI; k_1, k_2, and k_3 are random numbers from the range [0, 1]; and E and $F (E \neq F)$ are random numbers from the range [1, NP].

In the above updating rules, the first component $\Delta X_{ij}(t)$ reflects the influence of EI playing on the current learning of the learner. In the second component, instead of learning from other random learners, the learner learns from the difference between two random learners. In the third component, the learner learns from the local teacher of his corresponding neighborhoods.

On the contrary, if $r \geq P_m$, the EI is also updated according to (5). Then, the learner can also be updated according to (6).

The pseudocode of learner phase in the proposed EI-TLBO algorithm is given in Algorithm 2.

3) Pseudocode of EI-TLBO: The EI-TLBO algorithm has better exploration ability. On the one hand, EI can help the learner to make accurate judgment on the behavior in future. On the other hand, neighborhood information helps one to find a better solution around the neighbors of each learner. In addition, differential mutation helps one to maintain the diversity of learners during the whole learning process. Hence, the possibility of aggregation toward local optima is decreased. The pseudocode for the EI-TLBO algorithm can be generally described as in Algorithm 3, according to the above analysis.



$f1$ Sphere (a) $f2$ Quadric (b) $f6$ Ackley (c) $f7$ Rastrigin (d)

Fig. 2. Convergence curves with different u. (a) f_1 Sphere. (b) f_2 Quadric. (c) f_6 Ackley. (d) f_7 Rastrigin.

Algorithm 3 EI-TLBO( )

Begin
Initialize the class with *NP* learners with D dimensional variables
Evaluate all learners X
Determine the neighborhood of every learner
Denote $NTeacher_i$ and $NMean_i$ from the neighborhood of each learner X_i;
while(stopping condition not met)
Execute **Algorithm 1: Teaching ( );** % Teacher phase
Evaluate all learners $newX$;
Accept $newX_i$ if $newX_i$ is better than X_i for each learner X_i
Execute **Algorithm 2: Learning ( );** % Learner phase
Evaluate all learners $newX$;
Accept $newX_i$ if $newX_i$ is better than X_i for each learner X_i
endwhile
end

V. Experiments and Testing Results

To evaluate the efficiency of EI-TLBO, 46 benchmark functions, including ten widely used common benchmark functions, eight rotated benchmark functions, and 28 shifted benchmark functions from CEC2013 are used in our testing experiments. Six other algorithms including *j*DE [44], differential evolution algorithm with strategy adaptation (SaDE) [45], fully informed particle swarm (FIPS) [46], PSO-FDR [47], TLBO [16], and elitist teaching-learning-based optimization (ETLBO) [23] are also tested and evaluated in the paper to compare with EI-TLBO.

A. Parameters of Algorithms

All experiments are simulated with MATLAB software on Windows 7 operating system. For maintaining fairness of the comparison and reducing statistical errors, all experimental results are obtained from 50 independent runs. The termination condition of all algorithms is the maximal function evaluations (FEs) (150 000).

For EI-TLBO algorithm, three parameters (the weighting factor u, the mutation scaling factor F, and the probability P_m) need to be considered. By testing on various kinds of functions, the weighting factor u is empirically set to be 0.3 (see Fig. 2) and the probability P_m is empirically set to be 0.9. Considering the parameter values of the DE algorithm, the mutation scaling factor F is set to be 0.5. For ETLBO, elite size is set to 2. The parameters of other algorithms are the same as it is used in the corresponding reference.

TABLE I
18 Benchmark Functions

Function	Range	Optima	Accuracy
f_1 Sphere	[-100,100]	0	1E-8
f_2 Quadric	[-100,100]	0	1E-8
f_3 Sum Square	[-10,10]	0	1E-8
f_4 Zakharov	[-10,10]	0	1E-8
f_5 Rosenbrock	[-2.048,2.048]	0	5
f_6 Ackley	[-32.768,32.768]	0	1E-6
f_7 Rastrigin	[-5.12,5.12]	0	10
f_8 Weierstrass	[-0.5,0.5]	0	5
f_9 Griewank	[-600,600]	0	0.05
f_{10} Schwefel	[-500,500]	0	6E+2
f_{11} Rotated Sum Square	[-10,10]	0	1E-8
f_{12} Rotated Zakharov	[-10,10]	0	1E-8
f_{13} Rotated Rosenbrock	[-2.048,2.048]	0	10
f_{14} Rotated Ackley	[-32.768,32.768]	0	2
f_{15} Rotated Rastrigin	[-5.12,5.12]	0	10
f_{16} Rotated Weierstrass	[-0.5,0.5]	0	10
f_{17} Rotated Griewank	[-600,600]	0	0.2
f_{18} Rotated Schwefel	[-500,500]	0	4E+3

B. Performance on Common Benchmark Functions

1) Common Benchmark Functions: The 18 common benchmark functions are shown in Table I. Among these 18 benchmark functions, the first five are unimodal functions, the next five are multimodal functions, and the last eight functions are the rotated versions of f_3–f_{10}, respectively. The range of variables and theory optima of all functions are also given in Table I.

2) Solution Accuracy Comparisons of Different Algorithms: The merits in terms of mean and standard deviation of the best solutions for 30-D (D is the number of dimensions of functions) functions on 18 common benchmark functions with 50 independent runs are displayed in Table II. In addition, the difference of the merits with EI-TLBO and six other algorithms are tested with statistics method, pairwise t-tests with a significance level $\alpha = 0.05$ is used in the paper. "+" marks in Table II indicate that EI-TLBO statistically outperforms the compared algorithm, "−" marks indicate that EI-TLBO is statistically outperformed by the compared algorithm, and "=" marks indicate that there is no significant difference between EI-TLBO and the compared algorithms in the statistical sense. All marks are labeled after the results of the compared algorithms and the statistical results of pairwise t-tests are also given in the bottom row of Table II. The best values among all algorithms are shown in bold.

The comparisons in Table II show that EI-TLBO has shown the highest accuracy performance on 8 out of 18 functions (f_1–f_5, f_{11}, f_{12}, f_{15}). In particular, for unimodal problems, the

TABLE II
RESULT FOR COMMON BENCHMARK FUNCTIONS

NO.	Result	jDE	SaDE	FIPS	FDR-PSO	TLBO	ETLBO	EI-TLBO
f_1	Mean	1.14e-060 +	7.47e-058 +	1.12e-006 +	9.95e-072 +	1.20e-267 +	2.08e-229 +	**0.00e+00**
	Std	1.38e-060	1.30e-057	7.48e-007	1.23e-071	0.00e+000	0.00e+000	**0.00e+00**
f_2	Mean	1.19e-003 +	1.80e-003 +	2.12e+002 +	1.36e-004 +	1.46e-058 +	1.21e-056 +	**6.61e-101**
	Std	9.37e-004	1.76e-003	3.68e+001	7.64e-005	2.48e-058	2.64e-056	**1.47e-100**
f_3	Mean	4.36e-061 +	3.26e-058 +	1.19e-007 +	1.81e-071 +	1.96e-268 +	1.22e-229 +	**0.00e+00**
	Std	8.58e-061	7.28e-058	2.57e-008	2.37e-071	0.00e+000	0.00e+000	**0.00e+00**
f_4	Mean	3.87e-008 +	3.65e-009 +	8.98e-001 +	3.07e-008 +	2.40e-036 +	1.10e-035 +	**4.10e-83**
	Std	5.61e-008	4.06e-009	2.51e-001	3.79e-008	3.36e-036	1.17e-035	**5.02e-83**
f_5	Mean	9.72e+000 +	2.37e+001 +	2.56e+001 +	1.32e+001 +	1.81e+001 +	1.66e+001 +	**6.17e-01**
	Std	7.15e-001	8.88e-001	1.37e-001	1.16e+000	1.04e+000	6.90e-001	**3.70e-01**
f_6	Mean	4.97e-015 +	3.55e-015 +	2.27e-004 +	9.24e-015 +	**3.55e-015** =	**3.55e-015** =	**3.55e-15**
	Std	1.95e-015	0.00e+000	5.71e-005	4.77e-015	**0.00e+000**	**0.00e+000**	**0.00e+00**
f_7	Mean	**0.00e+000** =	**0.00e+000** =	4.99e+001 +	3.34e+001 +	8.16e+000 +	9.55e+000 +	**0.00e+00**
	Std	**0.00e+000**	**0.00e+000**	1.06e+001	1.15e+001	4.69e+000	6.62e+000	**0.00e+00**
f_8	Mean	3.02e-004 +	**0.00e+000** =	1.33e-002 +	8.58e-004 +	**0.00e+000** =	**0.00e+000** =	**0.00e+00**
	Std	6.75e-004	**0.00e+000**	1.91e-003	1.83e-003	**0.00e+000**	**0.00e+000**	**0.00e+00**
f_9	Mean	**0.00e+000** =	**0.00e+000** =	1.41e-004 +	1.23e-002 +	**0.00e+000** =	**0.00e+000** =	**0.00e+00**
	Std	**0.00e+000**	**0.00e+000**	1.25e-004	8.17e-003	**0.00e+000**	**0.00e+000**	**0.00e+00**
f_{10}	Mean	**3.82e-004** -	**3.82e-004** -	3.31e+003 -	3.03e+003 -	4.02e+003 +	3.82e+003 +	3.63e+03
	Std	**0.00e+000**	**0.00e+000**	6.56e+002	8.32e+002	7.94e+002	2.21e+002	6.99e+02
f_{11}	Mean	1.25e-027 +	2.12e-016 +	4.90e-005 +	2.04e-038 +	9.28e-257 +	4.91e-221 +	**0.00e+00**
	Std	2.80e-027	3.13e-016	6.90e-005	4.57e-038	0.00e+000	0.00e+000	**0.00e+00**
f_{12}	Mean	1.78e-008 +	5.66e-008 +	1.05e+000 +	6.83e-009 +	1.90e-036 +	1.56e-035 +	**1.30e-83**
	Std	2.67e-008	3.54e-008	1.76e-001	1.07e-008	3.54e-036	1.94e-035	**2.11e-83**
f_{13}	Mean	**1.55e+001** -	4.73e+001 +	2.64e+001 +	4.90e+001 +	3.01e+001 +	4.60e+001 +	2.35e+01
	Std	**1.04e+000**	2.84e+001	3.37e-001	2.41e+001	2.48e+001	2.21e+001	2.52e+01
f_{14}	Mean	4.26e-015 +	**3.55e-015** =	2.40e-004 +	8.67e-001 +	**3.55e-015** =	**3.55e-015** =	**3.55e-15**
	Std	1.59e-015	**0.00e+000**	3.49e-005	8.63e-001	**0.00e+000**	**0.00e+000**	**0.00e+00**
f_{15}	Mean	3.47e+001 +	6.41e+001 +	1.06e+002 +	4.30e+001 +	8.95e+000 +	1.13e+001 +	**9.15e+00**
	Std	8.18e+000	4.62e+000	1.58e+001	1.40e+001	3.07e+000	3.83e+000	**1.25e+01**
f_{16}	Mean	6.35e-002 +	1.16e-002 +	5.81e-002 +	2.81e+000 +	**0.00e+000** =	**0.00e+000** =	**0.00e+00**
	Std	1.01e-001	2.59e-002	7.50e-003	1.25e+000	**0.00e+000**	**0.00e+000**	**0.00e+00**
f_{17}	Mean	**0.00e+000** =	**0.00e+000** =	7.77e-004 +	1.76e-002 +	7.70e-005 +	**0.00e+000** =	**0.00e+00**
	Std	**0.00e+000**	**0.00e+000**	1.56e-003	2.12e-002	1.72e-004	**0.00e+000**	**0.00e+00**
f_{18}	Mean	2.41e+003 +	**2.32e+003** -	3.59e+003 -	3.86e+003 -	3.56e+003 -	4.82e+003 +	4.02e+03
	Std	2.84e+002	**7.53e+002**	9.78e+002	4.83e+002	8.68e+002	7.58e+002	7.16e+02
+/=/-		13/3/2	11/5/2	16/0/2	16/0/2	12/5/1	12/6/0	---

performance of EI-TLBO outperforms all other algorithms. In addition, EI-TLBO also finds the global optimum for complex multimodal functions f_1, f_9, f_{11}, f_{16}, and f_{17}. *j*DE outperforms EI-TLBO on f_{10} and f_{13}, SaDE outperforms EI-TLBO on f_{10} and f_{18}, the merits of *j*DE are worse than those of EI-TLBO on all other 13 functions except f_7, f_9, f_{10}, f_{13}, and f_{17}. The performances of SaDE are worse than those of EI-TLBO on all other 11 functions except f_7–f_{10}, f_{14}, f_{17}, and f_{18}. Moreover,

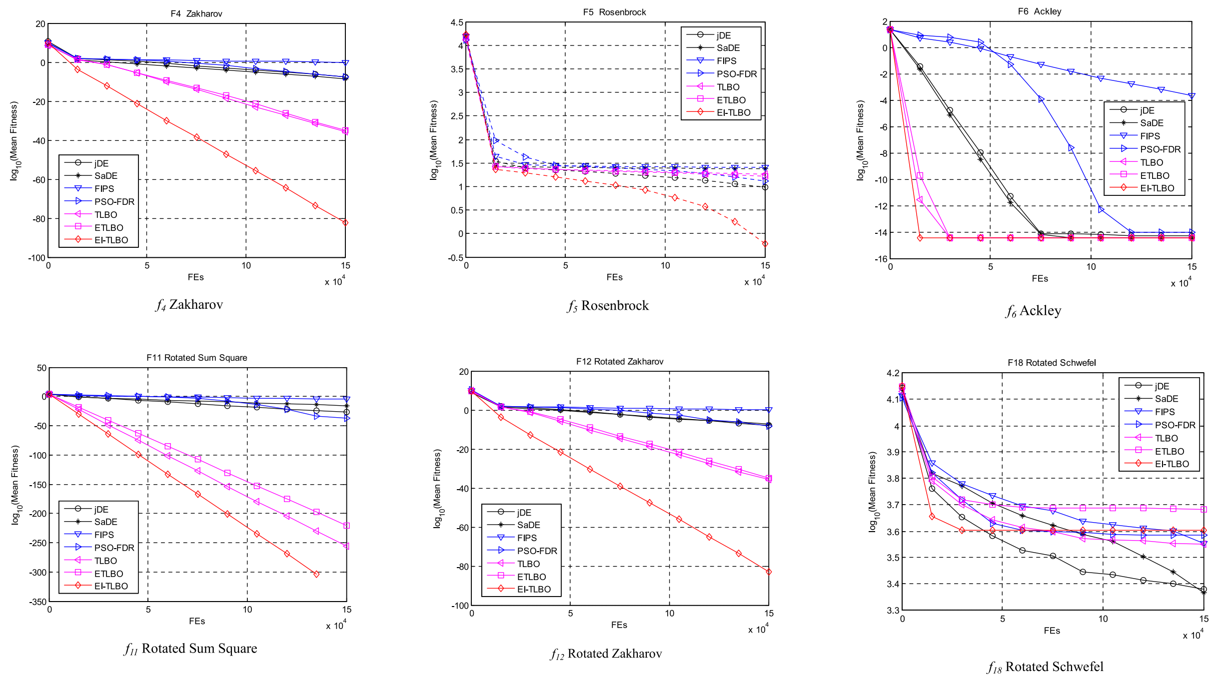


Fig. 3. Movements of average best solutions of different algorithms for some typical functions. (a) f_4 Zakharov. (b) f_5 Rosenbrock. (c) f_6 Ackley. (d) f_{11} Rotated sum square. (e) f_{12} Rotated Zakharov. (f) f_{18} Rotated Schwefel.

TABLE III
TABLE V COMPARISONS OF THE MEAN NUMBER OF FEs AND SUCCESSFUL RATIOS WITH ACCEPTABLE SOLUTIONS FOR 18 BENCHMARK FUNCTIONS

NO.	jDE		SaDE		FIPS		FDR-PSO		TLBO		ETLBO		EI-TLBO	
	MFEs	SR	MFEs	SR	MFEs	SR	MFEs	SR	MFEs	SR	MFEs	SR	MFEs	SR
f_1	30237	100.00	27391	100.00	--	0.00	78160	100.00	7093	100.00	8136	100.00	**2843**	100.00
f_2	--	0.00	--	0.00	--	0.00	--	0.00	29361	100.00	29740	100.00	**13539**	100.00
f_3	27490	100.00	25448	100.00	--	0.00	76467	100.00	6567	100.00	7683	100.00	**4173**	100.00
f_4	147280	60.00	141964	100.00	--	0.00	145625	60.00	52163	100.00	54435	100.00	**114**	100.00
f_5	--	0.00	--	0.00	--	0.00	--	0.00	--	0.00	--	0.00	**27591**	100.00
f_6	35527	100.00	33666	100.00	--	0.00	84010	100.00	8735	100.00	10120	100.00	**6617**	100.00
f_7	28943	100.00	50660	100.00	--	0.00	--	0.00	25741	80.00	15892	40.00	**4844**	100.00
f_8	16373	100.00	8116	100.00	27503	100.00	40434	100.00	2105	100.00	2324	100.00	**45**	80.00
f_9	15543	100.00	15212	100.00	84117	100.00	60320	100.00	3573	100.00	4236	100.00	**3111**	100.00
f_{10}	21173	100.00	26652	100.00	--	0.00	--	0.00	--	0.00	--	0.00	**19665**	60.00
f_{11}	45876	100.00	61515	100.00	--	0.00	86021	100.00	6785	100.00	7982	100.00	**4327**	100.00
f_{12}	144858	60.00	--	0.00	--	0.00	140672	80.00	51495	100.00	55496	100.00	**3720**	100.00
f_{13}	--	0.00	--	0.00	--	0.00	--	0.00	--	0.00	--	0.00	**43429**	80.00
f_{14}	8902	100.00	8816	100.00	37363	100.00	50388	100.00	2192	100.00	2474	100.00	**125**	100.00
f_{15}	--	0.00	--	0.00	--	0.00	--	0.00	32181	60.00	61036	60.00	**5758**	60.00
f_{16}	26895	100.00	7774	100.00	17462	100.00	35683	100.00	1586	100.00	1784	100.00	**340**	80.00
f_{17}	14276	100.00	14091	100.00	75109	100.00	58442	100.00	3601	100.00	3975	100.00	**3285**	100.00
f_{18}	39043	100.00	83110	100.00	69144	40.00	41595	40.00	59993	80.00	**31605**	20.00	--	0.00

EI-TLBO outperforms FIPS and fitness-distance ratio based particle swarm optimization (FDR-PSO) on almost all functions except f_{10} and f_{18}. EI-TLBO outperforms TLBO on 12 out of 18 functions, and outperforms ETLBO on 12 out of 18 functions, although EI-TLBO and TLBO perform well on f_6, f_8, f_9, f_{14}, and f_{16}, and EI-TLBO and ETLBO perform well on $f_6, f_8, f_9, f_{14}, f_{16}$, and f_{17}. It can be concluded that EI-TLBO has shown the most competitive overall performance

TABLE IV
RESULT FOR SHIFTED BENCHMARK FUNCTIONS

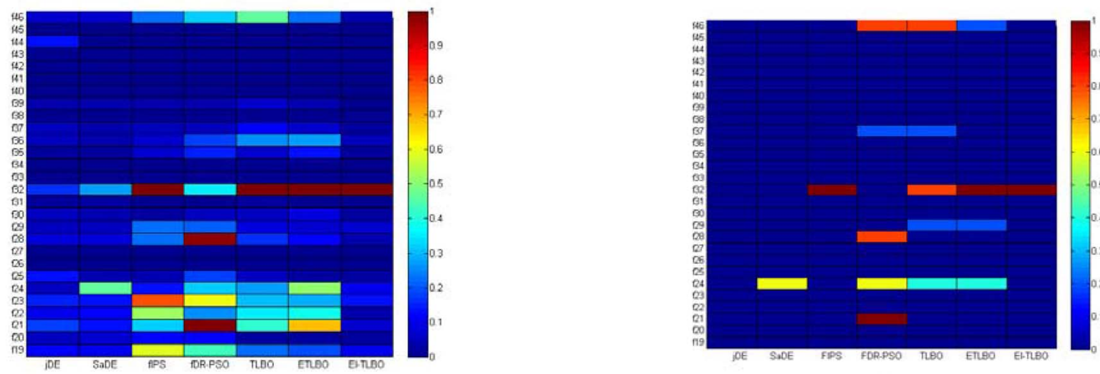
Fun	Result	jDE		SaDE		FIPS		FDR-PSO		TLBO		ETLBO		EI-TLBO
f_{19}	Mean	**0.00e+00**	-	**0.00e+00**	-	1.16e-06	+	2.27e-13	-	5.64e-12	+	4.64e-12	+	2.73e-13
	Std	**0.00e+00**		**0.00e+00**		4.43e-07		0.00e+00		7.40e-12		4.97e-12		1.02e-13
f_{20}	Mean	8.51e+05	+	4.45e+05	+	1.65e+07	+	2.26e+06	+	6.55e+05	+	1.15e+06	+	**2.02e+05**
	Std	8.00e+05		1.06e+05		4.51e+06		1.04e+06		3.04e+05		7.12e+05		**1.01e+05**
f_{21}	Mean	5.89e+06	+	2.60e+06	+	3.49e+07	+	2.24e+10	+	1.20e+08	+	2.35e+08	+	**3.84e+06**
	Std	9.25e+06		4.00e+06		1.67e+07		1.60e+10		4.95e+07		1.60e+08		**3.40e+06**
f_{22}	Mean	**5.80e+02**	-	1.87e+03	+	2.64e+04	+	9.75e+03	+	1.14e+04	+	1.36e+04	+	1.25e+03
	Std	**2.62e+02**		8.73e+02		1.41e+03		2.92e+03		2.58e+03		4.18e+03		7.48e+02
f_{23}	Mean	1.14e-13	-	**4.55e-14**	-	3.76e-04	+	2.27e-13	-	2.64e-12	+	2.80e-12	+	4.09e-13
	Std	0.00e+00		**6.23e-14**		7.84e-05		0.00e+00		1.05e-12		1.60e-12		1.30e-13
f_{24}	Mean	2.09e+01	+	4.76e+01	+	2.54e+01	+	6.78e+01	+	5.86e+01	+	4.09e+01	+	**7.10e+00**
	Std	1.23e+00		2.82e+01		1.11e+00		5.21e+01		4.85e+01		2.94e+01		**3.36e+00**
f_{25}	Mean	3.35e+01	-	**7.91e+00**	-	2.62e+01	-	6.63e+01	+	8.15e+01	+	6.61e+01	+	3.42e+01
	Std	2.52e+01		**4.39e+00**		8.55e+00		4.60e+01		3.45e+01		2.50e+01		1.50e+01
f_{26}	Mean	2.10e+01	=	2.10e+01	=	2.09e+01	-	**2.09e+01**	-	2.10e+01	=	2.10e+01	=	2.10e+01
	Std	5.31e-02		7.11e-02		8.81e-02		**7.07e-02**		6.33e-02		8.46e-02		2.38e-02
f_{27}	Mean	3.03e+01	+	2.88e+01	+	2.91e+01	+	1.77e+01	+	3.16e+01	+	3.10e+01	+	**1.51e+01**
	Std	1.23e+00		1.30e+00		3.35e+00		3.01e+00		3.98e+00		4.64e+00		**1.44e+00**
f_{28}	Mean	5.32e-02	+	3.18e-01	+	2.34e+00	+	6.54e+01	+	2.33e-01	+	1.98e-01	+	**5.17e-02**
	Std	2.50e-02		8.35e-02		6.07e-01		1.54e+01		1.48e-01		9.40e-02		**1.07e-02**
f_{29}	Mean	2.27e-14	-	**0.00e+00**	-	8.42e+01	+	4.65e+01	-	1.36e+02	+	1.32e+02	+	7.62e+01
	Std	3.11e-14		**0.00e+00**		1.21e+01		1.92e+01		1.90e+01		2.08e+01		1.54e+01
f_{30}	Mean	**5.76e+01**	-	8.71e+01	+	1.77e+02	+	7.09e+01	-	1.29e+02	+	1.46e+02	+	8.52e+01
	Std	**1.30e+01**		2.53e+01		7.05e+00		1.68e+01		2.48e+01		1.85e+01		2.12e+01
f_{31}	Mean	**1.03e+02**	-	1.12e+02	-	1.84e+02	+	1.34e+02	-	2.11e+02	+	2.02e+02	+	1.59e+02
	Std	**1.20e+01**		1.75e+01		1.51e+01		2.43e+01		3.73e+01		3.12e+01		2.31e+01
f_{32}	Mean	**1.53e-01**	-	1.23e+02	-	4.70e+03	-	1.44e+03	-	3.26e+03	-	5.30e+03	-	6.96e+03
	Std	**1.44e-01**		6.13e+01		5.13e+02		1.59e+02		1.91e+03		1.61e+03		3.87e+02
f_{33}	Mean	5.90e+03	-	6.03e+03	-	7.04e+03	+	**4.05e+03**	-	7.31e+03	+	7.20e+03	+	6.76e+03
	Std	4.91e+02		3.21e+02		1.82e+02		**9.34e+02**		1.47e+02		3.55e+02		4.82e+02
f_{34}	Mean	2.60e+00	+	**2.30e+00**	-	2.72e+00	+	2.44e+00	+	2.68e+00	+	2.51e+00	+	2.42e+00
	Std	2.70e-01		**4.63e-01**		4.17e-01		1.80e-01		4.23e-01		4.83e-01		3.73e-01
f_{35}	Mean	**3.04e+01**	-	4.15e+01	-	1.87e+02	-	7.03e+01	-	1.25e+02	-	1.37e+02	-	2.14e+02
	Std	**3.01e-06**		1.10e+00		8.83e+00		2.98e+00		2.82e+01		2.82e+01		1.58e+01
f_{36}	Mean	**1.61e+02**	-	1.82e+02	-	2.15e+02	-	1.73e+02	-	2.45e+02	+	2.58e+02	+	2.40e+02
	Std	**1.38e+01**		8.12e+00		1.22e+01		3.20e+01		2.80e+01		1.73e+01		1.40e+01
f_{37}	Mean	**1.73e+00**	-	4.56e+00	-	1.41e+01	+	1.88e+02	+	3.37e+01	+	1.50e+01	+	8.50e+00
	Std	**1.67e-01**		5.93e-01		9.27e-01		4.10e+02		2.51e+01		4.56e+00		2.45e+00
f_{38}	Mean	1.28e+01	+	1.28e+01	+	1.29e+01	+	1.32e+01	+	1.25e+01	+	1.24e+01	+	**1.19e+01**
	Std	3.44e-01		2.36e-01		3.80e-01		1.67e+00		3.54e-01		3.12e-01		**3.40e-01**
f_{39}	Mean	**2.60e+02**	=	3.29e+02	+	2.95e+02	+	3.86e+02	+	3.17e+02	+	3.86e+02	+	**2.60e+02**
	Std	**5.48e+01**		6.42e+01		2.12e+01		7.86e+01		1.41e+02		7.86e+01		**5.48e+01**
f_{40}	Mean	**2.49e+02**	-	7.15e+02	-	3.84e+03	+	1.41e+03	-	2.03e+03	+	1.99e+03	+	1.66e+03
	Std	**6.03e+01**		4.94e+02		3.34e+02		4.64e+02		6.59e+02		5.63e+02		6.21e+02
f_{41}	Mean	6.65e+03	-	6.76e+03	-	7.41e+03	+	**4.67e+03**	-	7.16e+03	+	7.34e+03	+	7.09e+03
	Std	4.02e+02		2.22e+02		4.36e+02		**1.17e+03**		5.14e+02		4.15e+02		3.71e+02
f_{42}	Mean	2.82e+02	+	**2.13e+02**	-	2.77e+02	+	2.77e+02	+	2.71e+02	+	2.65e+02	+	2.61e+02
	Std	8.34e+00		**4.60e+00**		5.04e+00		8.38e+00		9.63e+00		9.91e+00		5.96e+00
f_{43}	Mean	2.94e+02	+	2.86e+02	-	**2.80e+02**	-	2.90e+02	+	2.86e+02	+	2.90e+02	+	2.88e+02
	Std	5.40e+00		1.37e+01		**6.68e+00**		1.05e+01		6.14e+01		6.86e+00		1.00e+01
f_{44}	Mean	3.05e+02	-	**2.00e+02**	-	2.30e+02	-	3.45e+02	+	2.62e+02	-	2.00e+02	-	3.18e+02
	Std	9.65e+01		**1.28e-02**		6.53e+01		6.45e+00		8.43e+01		1.38e-01		6.63e+01
f_{45}	Mean	1.12e+03	+	**4.61e+02**	-	9.96e+02	+	8.55e+02	+	9.26e+02	+	9.02e+02	+	7.98e+02
	Std	5.57e+01		**1.30e+02**		8.13e+01		1.42e+02		4.15e+01		1.11e+02		1.47e+02
f_{46}	Mean	3.00e+02	+	**3.00e+02**	=	3.00e+02	+	9.65e+02	+	1.24e+03	+	7.54e+02	+	**3.00e+02**
	Std	2.49e-13		**0.00e+00**		1.62e-02		5.37e+02		7.73e+02		1.02e+03		**0.00e+00**
+/=/-		11/2/15		9/2/17		21/0/7		16/0/12		24/1/3		24/1/3		---

TABLE V
TABLE V COMPARISONS OF THE MEAN NUMBER OF FEs AND SUCCESSFUL RATIOS WITH ACCEPTABLE SOLUTIONS FOR CEC2013 TEST SUITE

NO.	jDE		SaDE		FIPS		FDR-PSO		TLBO		ETLBO		EI-TLBO	
	MFEs	SR	MFEs	SR	MFEs	SR	MFEs	SR	MFEs	SR	MFEs	SR	MFEs	SR
f_{19}	16104	100	15530	100	85412	100	61968	100	33322	100	28333	100	**13653**	100
f_{20}	8538	100	11109	100	17979	100	16434	100	4403	100	3840	100	**2853**	100
f_{21}	25436	100	20273	100	47345	100	--	0	59678	100	98766	100	**10912**	100
f_{22}	13967	100	16628	100	76395	100	37154	100	51326	100	55373	100	**6280**	100
f_{23}	21479	100	19457	100	113880	100	87202	100	43937	100	42083	100	**19191**	100
f_{24}	**8202**	100	67285	40	18513	100	46123	40	39083	60	73887	60	13768	100
f_{25}	18975	100	6079	100	5488	100	26151	100	7832	100	4384	100	**2944**	100
f_{26}	183	100	167	100	231	100	184	100	190	100	154	100	**137**	100
f_{27}	76	100	117	100	119	100	72	100	75	100	**50**	100	**50**	100
f_{28}	13382	100	11977	100	32096	100	141868	20	22967	100	16024	100	**6334**	100
f_{29}	**6847**	100	8487	100	32726	100	29608	100	12671	80	10847	80	9915	100
f_{30}	7296	100	4852	100	4160	100	7924	100	8157	100	13525	100	**2755**	100
f_{31}	3762	100	2323	100	1824	100	**734**	100	2301	100	2204	100	879	100
f_{32}	24889	100	39131	100	--	0	49942	100	144945	20	--	0	**--**	0
f_{33}	165	100	129.20	100	181	100	179	100	171	100	232	100	**81**	100
f_{34}	**313**	100	391	100	772	100	784	100	447	100	511	100	519	100
f_{35}	4411	100	**4372**	100	10491	100	22007	100	9830	100	18362	100	4465	100
f_{36}	6796	100	**5822**	100	10355	100	26319	100	36417	100	38594	100	8987	100
f_{37}	6994	100	5889	100	6911	100	8768	80	16025	80	10819	100	**3147**	100
f_{38}	108	100	116	100	127	100	109	100	89	100	92	100	**72**	100
f_{39}	5156	100	6678	100	6239	100	5616	100	10213	100	6514	100	**2091**	100
f_{40}	365	100	199	100	324	100	264	100	616	100	**195**	100	268	100
f_{41}	158	100	555	100	158	100	377	100	**90**	100	396	100	194	100
f_{42}	198	100	322	100	341	100	151	100	183	100	146	100	**95**	100
f_{43}	**76**	100	450	100	272	100	174	100	146	100	92	100	112	100
f_{44}	18412	100	**668**	100	1552	100	2080	100	1208	100	842	100	2013	100
f_{45}	137	100	203	100	155	100	122	100	95	100	102	100	**78**	100
f_{46}	10803	100	10794	100	32764	100	45470	20	67664	20	32778	80	**5550**	100

on this test suite. On the one hand, EI-TLBO has maintained the merit of fast convergence feature of TLBO, which can be demonstrated by its performance on functions f_1–f_5; on the other hand, its performance on complex rotated functions has been enhanced, which can be demonstrated by its performance on f_{10}–f_{13} and f_{15}. The global performance is improved by the given method.

3) Comparisons on Convergence Speed and Success Ratios: The mean FEs that the algorithm reaches the acceptable accuracy would be more commonly used for testing the convergence speed of the algorithms than the running time. In addition, successful ratio (reaching acceptable accuracies) is also an important evaluated index of the algorithm. In our experiments, acceptable accuracies have been given in Table I. The mean FEs and successful ratios that various algorithms reach the prior given acceptable accuracies are shown in Table III. Fig. 3 presents the convergence curves of typical functions.



(a) (b)

Fig. 4. Overview of mean FEs and successful ratios that various algorithms reach prior given acceptable accuracies. Results are normalized in the range [0, 1], 0 means the best algorithm and 1 means the worst algorithm. (a) Comparison on the MFEs needed. (b) Comparison on SR obtained.

The comparisons in Table III show that EI-TLBO spends the smallest FEs when reaching acceptable accuracies on 17 of 18 test functions except function f_{18}. Table III and Fig. 3 display that EI-TLBO generally obtains a much higher convergence speed and successful ratio. The mark "–" in Table III means that the algorithm cannot converge to the acceptable solution in 50 runs.

C. Testing Experiments for Shifted Benchmark Functions

To further evaluate the improved performance of EI-TLBO algorithm, the 28 shifted functions are taken from the CEC2013 test suite [41]. In this paper, they are marked with f_{19}–f_{46}. The range of variables and the theory optima are same as used in [48]. The acceptable solutions are 0.01, 5e7, 5e8, 4e4, 0.01, 50, 150, 50, 50, 50, 150, 300, 400, 2000, 1e4, 5, 300, 300, 50, 50, 1e3, 1e4, 1e4, 400, 400, 400, 1800, and 500 for 28 functions, respectively.

In our experiments, the convergence accuracies obtained from various algorithms are compared. Here, if the ideal global optima is x and the best global optima obtained by the algorithm is y, then the convergence accuracy of the algorithm can be described as $|y-x|$. The mean value, standard deviation, and the rank of the convergence accuracy are shown in Table IV for all 28 shifted functions. The best solutions among all algorithms are marked in bold. In addition, pairwise t-tests with significance level $\alpha = 0.05$ are also evaluated and the statistical results of pairwise t-tests are also given in the last row of Table IV.

Similar to the results shown in Table IV, among the 28 functions, the proposed EI-TLBO obtained the best results on seven of them (f_{20}, f_{21}, f_{24}, f_{27}, f_{28}, f_{38}, and f_{39}). In comparison, *j*DE outperforms all other algorithms on ten of them (f_{22}, f_{30}, f_{31}, f_{32}, f_{35}, f_{36}, f_{37}, f_{39}, and f_{40}); SaDE outperforms all other algorithms on eight of them (f_{23}, f_{25}, f_{29}, f_{34}, f_{42}, f_{44}, f_{45}, and f_{46}). FIPS obtained the best results on f_{43}, and FDR-PSO outperforms all other algorithms on f_{33} and f_{41}. It can be concluded that the proposed EI-TLBO has still shown the most competitive performance in comparison with the TLBO variants, although it is outperformed by *j*DE and SaDE.

Furthermore, the comparisons of the mean FEs and successful ratios that various algorithms reach the prior given acceptable accuracies are shown in Table V and Fig. 4. The comparisons in Table V and Fig. 4 show that the convergence speed of EI-TLBO is high and its mean FEs are also small with the acceptable solutions.

VI. Conclusion

In this paper, we have presented an EI-TLBO, which uses EI and differential mutation to improve the exploration capability of learners and maintain the diversity of learners. The 46 benchmark functions are used to evaluate the performance of the given algorithm, and some reported typical algorithms are also simulated to compare with EI-TLBO. From the results of the first 18 of 48 chosen test problems, it has been shown that the EI-TLBO algorithm has the highest accuracy performance on 9 out of 18 benchmark functions and the best overall performance on 18 benchmark functions. For 28 shifted benchmark functions in CEC2013, EI-TLBO does not have the best performance when compared with DE and SaDE, but it is challenging and promising when compared with other algorithms. The analysis and experiments also show that the EI-TLBO algorithm has significantly improved the search ability of the original TLBO and is effective and promising for global optimization problems.

Further works could consider extending adaptive weighting factor u to the iterative process, thus making the algorithm more efficient. This could improve the efficiency by making full use of EI. Moreover, how to improve the performance of the proposed algorithm for shifted functions optimization is an important work in the next phase. Extending EI-TLBO to solve real-world problems is also a significant work in future.

References

[1] C. L. P. Chen, Y.-J. Liu, and G.-X. Wen, "Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 583–593, May 2014.

[2] Z. Li, S. Xiao, S. S. Ge, and H. Su, "Constrained multilegged robot system modeling and fuzzy control with uncertain kinematics and dynamics incorporating foot force optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 1, pp. 1–15, Jan. 2016.

[3] Y.-J. Liu, L. Tang, S. C. Tong, C. L. P. Chen, and D.-J. Li, "Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 165–176, Jan. 2015.
[4] C. L. P. Chen, G.-X. Wen, Y.-J. Liu, and F.-Y. Wang, "Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1217–1226, Jun. 2014.
[5] Y.-J. Liu, L. Tang, S. C. Tong, and C. L. P. Chen, "Adaptive NN controller design for a class of nonlinear MIMO discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1007–1018, May 2015.
[6] Z. Li, H. Xiao, C. Yang, and Y. Zhao, "Model predictive control of nonholonomic chained systems using general projection neural networks optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 10, pp. 1313–1321, Oct. 2015.
[7] K. Yang, Z. Li, and J. Ye, "Freely-drawn sketches interpretation using SVMs-chain modeling," *Eng. Appl. Artif. Intell.*, vol. 25, no. 2, pp. 392–403, Mar. 2012.
[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading, MA, USA: Addison-Wesley, 1989.
[9] L. J. Fogel, "Evolutionary programming in perspective: The top-down view," in *Computational Intelligence: Imitating Life*. Piscataway, NJ, USA: IEEE Press, 1994.
[10] A. Qing, "Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems," *IEEE Geosci. Remote Sens.*, vol. 44, no. 1, pp. 116–125, Jan. 2006.
[11] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.
[12] M. Yang, C. H. Li, Z. H. Cai, and J. Guan, "Differential evolution with auto-enhanced population diversity," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 302–315, Feb. 2015.
[13] X. M. Hu, J. Zhang, H. S.-H. Chuang, Y. Li, and O. Liu, "SamACO: Variable sampling ant colony optimization algorithm for continuous optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1555–1566, Dec. 2010.
[14] Z. G. Ren, A. M. Zhang, C. Y. Wen, and Z. R. Feng, "A scatter learning particle swarm optimization algorithm for multimodal problems," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1127–1140, Jul. 2014.
[15] W. Gao, L. Huang, S. Liu, and C. Dai, "Artificial bee colony algorithm based on information learning," *IEEE Trans. Cybern*., vol. 45, no. 12, pp. 2827–2839, Dec. 2015.
[16] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Design*, vol. 43, no. 3, pp. 303–315, Mar. 2011.
[17] D. G. Unger and A. Wandersman, "The importance of neighbors: The social, cognitive, and affective components of neighboring," *Amer. J. Commun. Psychol.*, vol. 13, no. 2, pp. 139–169, Apr. 1985.
[18] R. Kundu *et al.*, "Improved CMA-ES with memory based directed individual generation for real parameter optimization," in *Proc. IEEE CEC*, Cancún, Mexico, 2013, pp. 748–755.
[19] D. Y. Tang, Y. M. Cai, J. Zhao, and Y. Xue, "A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems," *Inf. Sci.*, vol. 289, no. 12, pp. 162–189, Dec. 2014.
[20] C. Anderson and D. W. McShea, "Individual versus social complexity, with particular reference to ant colonies," *Biol. Rev.*, vol. 76, no. 2, pp. 211–237, May 2001.
[21] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "Bare bones differential evolution," *Eur. J. Oper. Res.*, vol. 196, no. 1, pp. 128–139, 2009.
[22] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: An optimization method for continuous nonlinear large scale problems," *Inf. Sci.*, vol. 183, no. 1, pp. 1–15, Jan. 2012.
[23] R. V. Rao and V. Patel, "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems," *Int. J. Ind. Eng. Comput.*, vol. 3, no. 4, pp. 535–560, Apr. 2012.
[24] B. Amiri, "Application of teaching-learning-based optimization algorithm on cluster analysis," *J. Basic Appl. Sci. Res.*, vol. 2, no. 11, pp. 11795–11802, Nov. 2012.
[25] V. Toğan, "Design of planar steel frames using teaching-learning based optimization," *Eng. Struct.*, vol. 34, pp. 225–232, Jan. 2012.
[26] T. Niknam, F. Golestaneh, and M. S. Sadeghi, "θ-multiobjective teaching-learning-based optimization for dynamic economic emission dispatch," *IEEE Syst. J.*, vol. 6, no. 2, pp. 341–352, Jun. 2012.
[27] R. V. Rao and V. Patel, "Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm," *Appl. Math. Model.*, vol. 37, no. 3, pp. 1147–1162, Feb. 2013.
[28] R. V. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Sci. Iran.*, vol. 20, no. 3, pp. 710–720, Jun. 2013.
[29] S. O. Degertekin and M. S. Hayalioglu, "Sizing truss structures using teaching-learning-based optimization," *Comput. Struct.*, vol. 119, pp. 177–188, Apr. 2013.
[30] S. C. Satapathy, A. Naik, and K. Parvathi, "Weighted teaching-learning-based optimization for global function optimization," *Appl. Math.*, vol. 4, no. 3, pp. 429–439, Mar. 2013.
[31] F. Zou, L. Wang, X. H. Hei, D. B. Chen, and D. D. Yang, "Teaching-learning-based optimization with dynamic group strategy for global optimization," *Inf. Sci.*, vol. 273, no. 8, pp. 112–131, Jul. 2014.
[32] B. Mandal and P. K. Roy, "Multi-objective optimal power flow using quasi-oppositional teaching learning based optimization," *Appl. Soft Comput.*, vol. 21, no. 3, pp. 590–606, Aug. 2014.
[33] C. V. Camp and M. Farshchin, "Design of space trusses using modified teaching–learning based optimization," *Eng. Struct.*, vols. 62–63, pp. 87–97, Mar. 2014.
[34] L. Wang *et al.*, "An improved teaching-learning-based optimization with neighborhood search for applications of ANN," *Neurocomputing*, vol. 143, no. 3, pp. 231–247, Nov. 2014.
[35] F. Zou *et al.*, "Bare-bones teaching-learning-based optimization," *Sci. World J.*, vol. 2014, no. 4, pp. 1–17, 2014.
[36] M. Ghasemi *et al.*, "Modified teaching learning algorithm and double differential evolution algorithm for optimal reactive power dispatch problem: A comparative study," *Inf. Sci.*, vol. 278, no. 10, pp. 231–249, Sep. 2014.
[37] Y.-H. Cheng, "Estimation of teaching-learning-based optimization primer design using regression analysis for different melting temperature calculations," *IEEE Trans. Nanobiosci.*, vol. 14, no. 1, pp. 3–12, Jan. 2015.
[38] M. Ghasemi, M. Taghizadeh, S. Ghavidel, J. Aghaei, and A. Abbasian, "Solving optimal reactive power dispatch problem using a novel teaching–learning-based optimization algorithm," *Eng. Appl. Artif. Intell.*, vol. 39, no. 3, pp. 100–108, Mar. 2015.
[39] Y. Xu, L. Wang, S.-Y. Wang, and M. Liu, "An effective teaching–learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time," *Neurocomputing*, vol. 148, no. 1, pp. 260–268, Jan. 2015.
[40] R. Kadambur and P. Kotecha, "Multi-level production planning in a petrochemical industry using elitist teaching–learning-based-optimization," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 628–641, Jan. 2015.
[41] X. Hu and R. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 2. Honolulu, HI, USA, May 2002, pp. 1677–1681.
[42] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "Using the ring neighborhood topology with self-adaptive differential evolution," in *Advances in Natural Computation*. Berlin, Germany: Springer, 2006, pp. 976–979.
[43] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.
[44] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.
[45] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.
[46] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.
[47] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in *Proc. Swarm Intell. Symp.*, Indianapolis, IN, USA, 2003, pp. 174–181.
[48] J. Yu, S. J. Wang, and L. F. Xi, "Evolving artificial neural networks using an improved PSO and DPSO," *Neurocomputing*, vol. 71, nos. 4–6, pp. 1054–1060, Jan. 2008.

Zhuo Wang (M'15) received the B.E. degree in automation from Beihang University, Beijing, China, in 2006 and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Chicago, Chicago, IL, USA, in 2013.

He was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada, from 2013 to 2014. Since 2014, he has been a Research Assistant Professor and a Post-Doctoral Fellow with Fok Ying Tung Graduate School, Hong Kong University of Science and Technology, Hong Kong. His current research interests include data-based system analysis and control, nonlinear systems adaptive control, trajectory tracking control, neural-network-based system analysis and control, and event-triggered control.

Renquan Lu (M'08) received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2004.

He is currently a Full Professor with the Institute of Information and Control, Hangzhou Dianzi University, Hangzhou. He has published over 30 journal papers in the fields of robust control and complex systems. His current research interests include robust control singular systems, and complex systems.

Debao Chen received the Ph.D. degree in pattern recognition and intelligent system from the School of Computer Science, Nanjing University of Science and Technology, Nanjing, China, in 2008.

He is currently a Full Professor with Huaibei Normal University, Huaibei, China. His current research interests include evolutionary computation, global optimization, multiobjective optimization, and neural networks.

Feng Zou received the Ph.D. degree in control theory and control engineering from the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China, in 2015.

He is an Associate Professor with Huaibei Normal University, Huaibei, China. His current research interests include evolutionary algorithms, swarm intelligence, and multiobjective optimization.