Mathematical Biosciences and Engineering

MBE, 20(11): 20317–20344.
DOI: 10.3934/mbe.2023899
Received: 20 August 2023
Revised: 20 October 2023
Accepted: 02 November 2023
Published: 08 November 2023

http://www.aimspress.com/journal/MBE

Research article

Optimizing boiler combustion parameters based on evolution teaching-learning-based optimization algorithm for reducing NO_x emission concentration

Yunpeng Ma[1], Shilin Liu[1], Shan Gao[1,*], Chenheng Xu[2] and Wenbo Guo[1]

[1] School of Information Engineering, Tianjin University of Commerce, Tianjin, China
[2] School of Economics, Tianjin University of Commerce, Tianjin, China

* **Correspondence:** Email: gaoshan@tjcu.edu.cn; Tel: +8613802137543.

Abstract: How to reduce a boiler's NO_x emission concentration is an urgent problem for thermal power plants. Therefore, in this paper, we combine an evolution teaching-learning-based optimization algorithm with extreme learning machine to optimize a boiler's combustion parameters for reducing NO_x emission concentration. Evolution teaching-learning-based optimization algorithm (ETLBO) is a variant of conventional teaching-learning-based optimization algorithm, which uses a chaotic mapping function to initialize individuals' positions and employs the idea of genetic evolution into the learner phase. To verify the effectiveness of ETLBO, 20 IEEE congress on Evolutionary Computation benchmark test functions are applied to test its convergence speed and convergence accuracy. Experimental results reveal that ETLBO shows the best convergence accuracy on most functions compared to other state-of-the-art optimization algorithms. In addition, the ETLBO is used to reduce boilers' NO_x emissions by optimizing combustion parameters, such as coal supply amount and the air valve. Result shows that ETLBO is well-suited to solve the boiler combustion optimization problem.

Keywords: optimization; model; extreme learning machine; teaching-learning-based optimization algorithm; evolution computation; boiler combustion optimization

1. Introduction

1.1. Background and present situation

With the rapid development of science and technology, more and more engineering problems can be regarded as strict optimization problems. For these optimization problems, the preliminary work mainly focuses on various mathematical techniques, but these methods probably cannot find the global optimal solution effectively. On the contrary, many intelligent optimization algorithms inspired by natural phenomena have been developed and widely used in various scientific and technological fields instead of traditional optimization algorithms. These intelligent optimization techniques show ideal results in solving complex engineering problems, such as structural design problems, multi-channel steering operation problems and grinding operation problems, so intelligent optimization techniques have attracted the attention of many scholars [1]. Until now, outstanding heuristic intelligence optimization algorithms [2–4] have been proposed to solve complex problems, which are shown in Table 1. All these artificial intelligence optimization algorithms have been successfully applied to various optimization problems, and the effectiveness of these intelligent algorithms has been proved.

Table 1. Outstanding heuristic intelligence optimization algorithms.

Abbreviation	Full name
PSO	Particle Swarm Optimization [5]
ACO	Ant Colony Optimization [6]
SFLA	Shuffled Frog Leaping Algorithm [7]
ABC	Artificial Bee Colony [8]
AFSO	Artificial Fish Swarm Optimization [9]
GWO	Grey Wolf Optimization [10]
BFO	Bacteria Foraging Optimization [11]
WOA	Whale Optimization Algorithm [12]
SO	Snake Optimizer [13]
GSA	Gravitational Search Algorithm [14]
GA	Genetic Algorithm [15]
ALO	Ant Lion Optimizer [16]
DA	Dragonfly Algorithm [17]
MFO	Moth-Flame Optimization [18]
SCA	Sine Cosine Algorithm [19]
TLBO	Teaching-Learning-Based Optimization [20]
ETLBO	Evolution teaching-learning-based optimization [our method]

In 2010, Indian scholar Rao et al. [20] proposed a swarm intelligence algorithm—the teaching-learning-based optimization (TLBO) algorithm, which is proposed and inspired by the teaching-learning phenomenon of a class. Because of its own advantages, it has inspired a wide range of studies and applications. The TLBO algorithm has been successfully applied to function optimization problems, engineering optimization problems and some other practical applications [21–23].

1.2. The research state of TLBO algorithm

The TLBO algorithm will be studied in this paper. The optimization idea of this algorithm regards the population as a class, in which the individual with the best fitness is the teacher. The teacher can improve the average score of the whole class through teaching activities, so as to realize the optimization evolution of the whole population. Students communicate with each other through a certain mechanism to maintain the diversity of the population and avoid premature convergence of the algorithm. The principle of TLBO is simple and easy to understand, and requires few parameters to be set. TLBO has attracted the attention of many researchers at home and abroad since it was proposed. It has been successfully applied to the optimization of large-scale continuous nonlinear problems [24], identifying photovoltaic cell model parameters [25], optimizing distribution of local automatic voltage adjustment in distributed systems [26], data clustering [27], optimization of assembly sequence planning for industrial robots [28], the set alliance knapsack problem [29] and other problems.

However, the TLBO algorithm still has several shortcomings. For instance, the TLBO algorithm is accomplished in solving low-dimensional or high-dimensional uni-modal functions, but for multi-modal functions, it is easy to get trapped in a local optimum, which is caused by the update mechanism during the teaching phase. With the progress of iteration, the population individuals approach the optimal solution, causing the loss of population diversity. The convergence accuracy, convergence speed and arithmetic speed of the TLBO algorithm still needs to be further improved.

In recent years, domestic and foreign researchers have conducted extensive and in-depth studies on the issues of the TLBO algorithm mentioned above. Ghasemi et al. [30] introduced mutation operator to the learning phase of the TLBO algorithm to enhance the population diversity. Li et al. [31] introduced inertia weight and acceleration weight to the teaching phase and learning phase to improve its convergence speed and the quality of the solution. Wang et al. [32] designed a sub-population-based teaching phase to enhance particle diversity and improve the convergence speed of the algorithm. Yu et al. [33] introduced feedback, mutation and crossover operators from differential evolution and chaotic wave algorithms to the TLBO algorithm, respectively improving its development capability, the diversity of the population and its ability to escape local optima. Tsai [34] constructed a mutation strategy by randomly selecting the difference vector of two individuals as the third individual's mutation source. Rao and Patel [35] introduced the elite mechanism to the TLBO algorithm, improving its convergence accuracy. Zou et al. [36] introduced the dynamic grouping mechanism to the TLBO algorithm to enhance its global search capability. Chen et al. [37] introduced the local learning and self-learning mechanisms to the TLBO algorithm to enhance its search capability. Sultana and Roy [38] introduced the reverse learning and quasi-reverse learning mechanisms to improve its convergence speed and the quality of the solution. Zou et al. [39] solved global optimization problems by adding dynamic group strategy to the TLBO algorithm, thus improving its global search capability. Tuo et al. [40] combined the harmony search and TLBO algorithms to effectively solve complex high-dimensional optimization problems.

In order to further improve the performance of the TLBO algorithm, three improvement mechanisms are introduced: 1) the chaotic mapping function is used to initialize the population individuals to increase population diversity and enhance the global search capability. 2) In the "teaching phase", three parameters, namely inertia weight, acceleration coefficient and self-adaptive teaching factor, are introduced to improve the algorithm's arithmetic speed and the quality of the solution. 3) In the "learning phase", the idea of heredity is used to update the population. The latest

individuals are taken as the next iteration's population to maintain population diversity in the later stage of optimization and improve the global search capability. 20 IEEE congress on Evolutionary Computation (CEC) benchmark test functions are used to verify the performance of the proposed algorithm. Compared to several state-of-the-art algorithms, namely GA, ALO, DA, MFO, SCA and TLBO, the experimental results show that the proposed algorithm has excellent performance in terms of convergence accuracy and the global search capability.

1.3. Boiler combustion optimization by heuristic optimization algorithm

During the combustion process of a station boiler, large amounts of polluting gases are produced, such as NO_x, SO_2 and CO_2, which cause great harm to the human living environment. Simultaneously, a large amount of coal is consumed. Therefore, the realization of dynamic multi-objective optimal control of the boiler combustion process under variable load is an effective method to reduce environmental pollution and save coal resources, which is called the boiler combustion optimization problem. Therefore, the boiler combustion optimization problem can be classified into a class of variable load, multi-variable, constrained dynamic multi-objective optimization problems. In recent years, with the rapid development of artificial intelligence technology, many researchers tried to use machine learning and heuristic optimization algorithms to optimize the adjustable operating parameters of the boiler combustion process, to achieve the goal of improving the boiler thermal efficiency or reducing the emission concentration of polluting gases [41–45].

The power station boiler has the characteristics of nonlinearity, strong coupling and large lag, which make it difficult for the traditional optimization method to achieve the goal of energy saving and emission reduction of the boiler. However, the heuristic intelligent optimization algorithm can realize the optimization and adapt to the uncertainty in the optimization process in the absence of systematic accurate analytical expressions or mathematical models. Therefore, domestic and foreign scholars are keen to apply the heuristic intelligent optimization algorithm to solve the boiler combustion optimization problem [46–48]. Rahat et al. used the novel multi-target evolutionary algorithm and data-driven model to find the equilibrium relationship between the emission concentration of nitrogen oxides and the carbon content of fly ash, effectively solving the contradiction between boiler thermal efficiency and NO_x [49]. Reference [50] proposed a boiler combustion optimization algorithm based on big data driven case matching, using data mining technology to analyze the data in Supervisory Information System (SIS) and establish the combustion case database. Online optimization can match the real-time operation data of a distributed control system (DCS) and case database, finding the best operating parameters suitable for current working conditions, and realizing the online optimization of boiler combustion. Reference [51] used a deep neural network and multi-objective optimization algorithm to achieve multi-objective optimization of the boiler combustion process, effectively balancing the thermal efficiency and nitrogen oxide emission concentration. In this paper, the proposed ETLBO is combined with extreme learning machine to solve the boiler combustion problem for reducing NO_x emissions.

The contributions of this paper are summarized as follows:

1) A kind of evolutionary teaching-learning-based optimization algorithm (ETLBO) is proposed.

2) The proposed ETLBO is used to solve 20 benchmark testing functions.

3) The proposed ETLBO-ELM method is applied to optimize the adjustment operation parameters of a 330 MW circulation fluidized bed boiler for reducing NO_x emissions.

The structure of this paper is as follows: The basic TLBO algorithm is given in Section 2. The proposed ETLBO is given in Section 3. Section 4 shows the performance evaluation of the ETLBO. Section 5 shows the boiler combustion model and optimization. The conclusion of this paper is in Section 6.

2. Basic TLBO algorithm

The TLBO algorithm is inspired by the teaching-learning phenomenon. Students are regarded as population individuals, their grades in each subject are regarded as the solutions to be optimized, the number of subjects is the solution dimension and the best individual becomes the teacher. The core idea of the TLBO algorithm is to simulate the teaching-learning process of a class. First, the best individual in the population is selected as the teacher, who improves the students' grades through teaching, thus achieving the goal of improving the average grade of the class. Students can learn from each other by comparing their grades, complementing each other's strengths and making progress together. Therefore, the TLBO algorithm is divided into two phases: "teaching phase" and "learning phase".

2.1. Teaching phase

In an ideal situation, students can learn from the teacher's guidance and reach the same level as the teacher. However, due to the interaction of many other factors, the teacher can only help the students reach a certain level. This phenomenon can be expressed mathematically as follows:

$$\text{difference\_mean}_i = r_i(M_{teacher} - T_F M_i), \tag{1}$$

$$T_F = \text{round}[1 + \text{rand}(0,1)], \tag{2}$$

$$X_{new,i} = X_{old,i} + \text{difference\_mean}_i. \tag{3}$$

Formula (1) represents the difference between the best grade and average grade; r_i is a random number between 0 and 1; $M_{teacher}$ is the best individual, regarded as teacher; T_F is a teaching factor, which is shown in formula (2); M_i is the average score at the i-th iteration. In formula (3), $X_{old,i}$ is the old solution at the i-th iteration and $X_{new,i}$ is the new solution after updating.

2.2. Learning phase

Students learn from each other through mutual communication and learning to acquire knowledge. Students with lower grades learn from those with higher grades, complementing each other's strengths and making progress together. Two students, X_i and X_j, are randomly selected. For the minimum optimization problem, if X_i is better than X_j, then X_j learns from X_i; otherwise, if X_j is better than X_i, then X_i learns from X_j. This phenomenon can be expressed mathematically as follows:

$$X_{new,i} = X_{old,i} + r_i(X_i - X_j), \quad \text{if } f(X_i) < f(X_j), \tag{4}$$

$$X_{new,i} = X_{old,i} + r_i(X_j - X_i), \quad \text{if } f(X_i) \geq f(X_j). \tag{5}$$

As shown in formulas (4) and (5), if the updated $X_{new,i}$ is better than the old $X_{old,i}$, then $X_{new,i}$ is accepted; otherwise, $X_{old,i}$ remains unchanged.

The pseudo-code of the original TLBO is given in Algorithm 1.

Algorithm 1

1: Objective function $f(x),\ x_i(i = 1,2,\ldots,n)$

2: Initialize algorithm parameters.

3: Generate the initial population of individuals.

4: Evaluate the fitness of the population.

5: **While** the stopping criteria is not adequate **do**

6: **Teaching phase**

7: Select the best individual X_{best} in the current population.

8: Calculate the mean value X_{mean}.

9: **For** each student in population **do**

10: X_i learn from the X_{best} and produce new solution $X_{new,i}$ by using Eq (3).

11: Evaluate new solutions.

12: Update better solutions.

13: **End For**

14: **Learning phase**

15: **For** each student in population **do**

16: Randomly select a learner X_j.

17: **If** X_i is superior than X_j **then**

18: Produce new solution $X_{new,i}$ by using Eq (4).

19: **Else**

20: Produce new solution $X_{new,i}$ by using Eq (5).

21: **End If**

22: Evaluate new solutions.

23: Update better solutions.

24: **End For**

25: gen = gen + 1.

26: **End While**

27: Output the best solution found.

3. The proposed ETLBO algorithm

In order to improve the convergence accuracy and convergence speed of the original TLBO algorithm, a kind of ETLBO algorithm is proposed. The ETLBO algorithm adopts three adjustment mechanisms: First, the initialization of the population individuals is improved by applying chaotic mapping sequences, which can enhance the population diversity. Second, in the teaching phase, three coefficients are introduced to improve the convergence speed and the solution quality. Finally, in the learning phase, the idea of heredity is used to update the population, and all the latest individuals

are taken as the population for the next iteration, which helps to maintain the diversity of the population in the later stage of optimization and further improve the global search capability. The specific implementation process of the ETLBO algorithm is described in the following subsections.

3.1. Using chaos mapping to initialize individuals

In the original TLBO algorithm, the population individuals were initialized using pseudo-random sequences, which resulted in a high degree of uncertainty in the diversity of the population and could not guarantee that the solutions were uniformly distributed in the search space, ultimately leading to premature convergence of the algorithm. Since chaos mapping has the properties of traverseability, randomness and overall stability, using the chaotic sequence generated by the chaos mapping function as the initial position of the population individuals can enhance the uniformity of traversing the initial solution, thereby improving the diversity of the population and the global search capability. Therefore, this paper adopts logistic chaos mapping to initialize the population, and the standard logistic chaos mapping is:

$$Z_{k+1} = \mu Z_k(1 - Z_k), \tag{6}$$

$$Z_0 \notin \{0,0.25,0.5,0.75,1.0\},\ \mu \in [0,4].$$

In formula (6), μ is a random number between 0 and 4; Z_k is the k-th chaotic variable with the value range of [0,1].

After considering various factors, this paper sets $\mu = 4$. The specific formula is as follows:

$$Z_{k+1} = 4Z_k(1 - Z_k), \tag{7}$$

$$Z_0 \notin \{0,0.25,0.5,0.75,1.0\},\ \mu \in [0,4].$$

Finally, the formula for converting chaotic solutions into solutions in the solution space of practical problems is as follows:

$$pop_{new} = \frac{range}{2}(1 + pop_{old}) + lower. \tag{8}$$

In formula (8), pop_{old} is the population matrix before transformation; $range$ is a matrix consisting of 60×1 copies of $(ub - lb)$; $lower$ is a matrix consisting of 60×1 copies of lb; ub is the upper constraint; lb is the lower constraint; pop_{new} is the population matrix after transformation.

3.2. Teaching phase

In the original TLBO algorithm, the teaching coefficient T_F affects the search speed and search ability of the algorithm, and determines the change of the average value. Among them, a larger T_F value speeds up the search speed, but it also makes the algorithm easily get trapped into local optima. A smaller T_F value slows down the search speed, but it also makes the algorithm escape local optima. In addition, the teaching coefficient T_F is a random number with a value of 1 or 2, and students can only fully accept or fully reject the teacher's teaching, which is inconsistent with the actual situation. After

considering various situations, this paper proposes a composite individual update mechanism in the stage where the teacher teaches students, and three parameters are set, including inertia weight ω_i, acceleration coefficient ϕ_i and self-adaptive teaching coefficient T_F. The introduced inertia weight and acceleration coefficient can improve the search speed and the quality of the solution of the algorithm. The self-adaptive teaching coefficient is a monotonically decreasing function, which is related to the current number of iterations and the maximum number of iterations. The self-adaptive teaching coefficient can speed up the convergence speed of the algorithm in the early stage and slow down the convergence speed in the later stage to avoid getting trapped in local optimization, and the combination of the two can avoid premature convergence of the algorithm. The specific formula is as follows:

$$\omega_i = 1/(1 + \exp(-f(i)/a) \times t), \tag{9}$$

$$\phi_i = 1/\left(1 + \exp(-f(i)/a)\right)^t, \tag{10}$$

$$T_F = 1 + \cos(\pi t/2T), \tag{11}$$

$$X_{new,i} = \omega_i X_{old,i} + \phi_i(M_{teacher} - T_F M_i). \tag{12}$$

In formula (9), ω_i is the inertia weight, which affects $X_{old,i}$. In formula (10), ϕ_i is the acceleration coefficient, which improves the search speed of the "teaching phase". In formula (11), T_F is the self-adaptive teaching factor, which determines the degree of early maturity of the algorithm. $f(i)$ is the fitness value of the i-th student, which is shown in formulas (9) and (10). a is the maximum fitness value in the iteration, which is shown in formulas (9) and (10). t is the current number of iterations, which is shown in formulas (9)–(11). T is the maximum number of iterations, which is shown in formula (11). In formula (12), $M_{teacher}$ is the best individual in the population, that is, the teacher. M_i is the average score at the i-th iteration. As shown in formula (12), if the updated $X_{new,i}$ is better than the old $X_{old,i}$, then $X_{new,i}$ is accepted; otherwise, $X_{old,i}$ remains unchanged.

3.3. Learning phase

During the learning process of the original TLBO algorithm, the population of the next iteration is generated through mutual communication and learning among students. The main idea is to randomly select two students and let the one with poor performance learn from the one with good performance. However, this method reduces the diversity of the population in the later optimization stages, which can easily lead to getting trapped in local optimization. As the idea of heredity is a type of search technique based on self-adaptive probability, it increases the flexibility of the search process. Although this probability feature may produce some low-fitness individuals, more excellent individuals will be generated as the evolution process continues, gradually making the population evolve to a state containing an approximate optimal solution. Moreover, the idea of heredity is scalable and easy to combine with other algorithms to generate hybrid algorithms that integrate the advantages of both. Based on the learning process of the original TLBO algorithm, this paper proposes adopting the idea of heredity to update the population, where all the latest individuals generated through crossover and mutation operations are used as the population of the next iteration. This method helps to maintain the diversity of the population in the later optimization stages and further improve the global search capability. The specific computation steps of ETLBO are described below and the

flowchart of the ETLBO algorithm is presented in Figure 1.

Step-1: The mutual learning among students is conducted according to the "learning phase" of the original TLBO algorithm, where the students with poor performance learn from those with good performance. This method satisfies the following conditions: if the updated $X_{new,i}$ is better than the old $X_{old,i}$, then $X_{new,i}$ is accepted; otherwise, $X_{old,i}$ remains unchanged.

Step-2: Sort all individuals in ascending order based on their fitness values and divide them into two groups, A and B, according to certain rules.

Group A: Individuals ranked 1st, 3rd, 5th, 7th, ..., etc.

Group B: Individuals ranked 2nd, 4th, 6th, 8th, ..., etc.

Step-3: Crossover operation: The first half of individuals in groups A and B are crossed over according to certain rules, and the resulting offspring individuals replace the second half of individuals with lower rankings. The specific formula is as follows:

$$X_{new,j} = \begin{cases} 0.5 \times [(1+\beta) \times A_i + (1-\beta) \times B_i] \\ 0.5 \times [(1-\beta) \times A_i + (1+\beta) \times B_i] \end{cases}, \quad \begin{cases} i = 1,2,\dots,\frac{N}{4} \\ j = \frac{N}{4}+1, \frac{N}{4}+2, \dots, \frac{N}{2} \end{cases}, \tag{13}$$

$$\beta = \begin{cases} (r+2)^{\frac{1}{1+\eta}}, & r \le 0.5 \\ \left(\frac{1}{2-2r}\right)^{\frac{1}{1+\eta}}, & r > 0.5 \end{cases}. \tag{14}$$

In formula (13), A_i is the i-th student in Group A, B_i is the i-th student in Group B and N is the population size. In formula (14), β is a balancing parameter, r is a random number between 0 and 1 and η is a custom parameter value where the larger the value, the closer the offspring individuals are to their parents.

Step-4: Mutation operation: All students are mutated one by one according to the mutation probability P_m. For a property value of an individual, if the random number $r \in [0,1] < P_m$, a mutation operation is performed, that is, the property value is inverted. The specific formula is as follows:

$$X'_{i,j} = ub + lb - X_{i,j}, \quad \begin{cases} i = 1,2,\dots,N \\ j = 1,2,\dots,d \end{cases}. \tag{15}$$

In formula (15), $X_{i,j}$ is the j-th property value of the i-th individual, ub is the upper constraint, lb is the lower constraint, N is the population size and d is the population dimension.

Step-5: Take all the latest individuals generated by crossover and mutation operations as the population for the next iteration.

After calculation, the computational complexity of ETLBO is $O(\text{Max\_iter} \times \text{pop\_num} \times \text{dim})$. Max_iter is the maximum number of iterations, pop_num is the population size and dim is the population dimension.

The pseudo-code of the proposed ETLBO is given in Algorithm 2.

Algorithm 2

1: Objective function $f(x)$, $x_i(i = 1,2,\ldots,n)$
2: Initialize algorithm parameters.
3: Use logistic chaos mapping to generate the initial population of individuals.
4: Evaluate the fitness of the population.
5: **While** the stopping criteria is not adequate **do**
6: **Teaching phase**
7: Select the best individual X_{best} in the current population.
8: Calculation the mean value X_{mean}.
9: **For** each student in population **do**
10: X_i learn from the X_{best} and produce new solution $X_{new,i}$ by using Eq (12).
11: Evaluate new solutions.
12: Update better solutions.
13: **End For**
14: **Learning phase**
15: **For** each student in population **do**
16: Randomly select a learner X_j.
17: **If** X_i is superior to X_j **then**
18: Produce new solution $X_{new,i}$ by using $X_{new,i} = X_{old,i} + r(X_i - X_j)$.
19: **Else**
20: Produce new solution $X_{new,i}$ by using $X_{new,i} = X_{old,i} + r(X_j - X_i)$.
21: **End If**
22: Evaluate new solutions.
23: Update better solutions.
24: **End For**
25: **Crossover and Mutation**
26: Sort by fitness value.
27: Divide the students into two groups according to the Step-2.
28: **For** each student in the first half of each group **do**
29: Perform crossover operation.
30: Produce new solution $X_{new,i}$ by using Eq (13).
31: **End For**
32: Replace the lower-ranked students with the offspring students obtained.
33: **For** each student in population **do**
34: Perform mutation operation according to the mutation probability.
35: Produce new solution $X_{new,i}$ by using Eq (15).
36: **End For**
37: Take all the latest students as the population for the next iteration.
38: gen=gen+1.
39: **End While**
40: Output the best solution found.

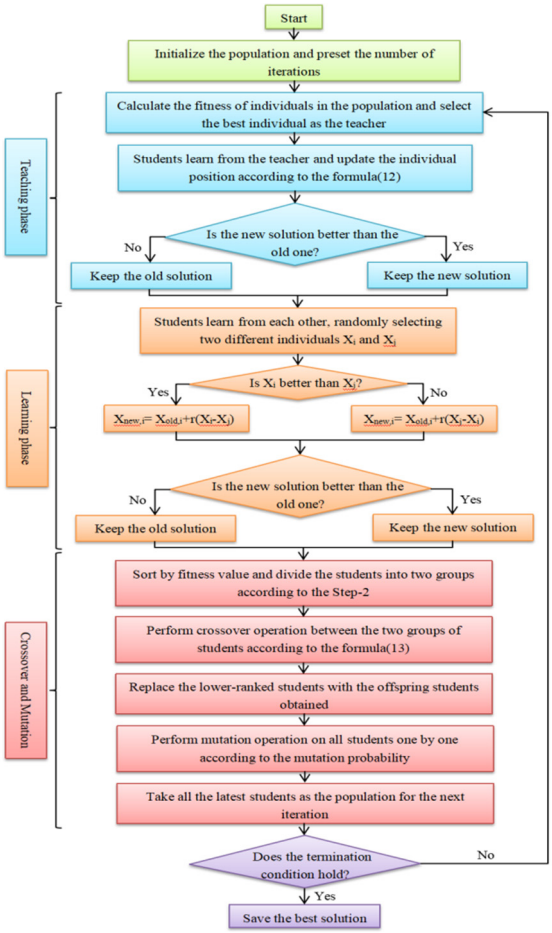

Figure 1. The flowchart of the ETLBO algorithm.

4. Performance testing of the ETLBO algorithm

In this section, 20 benchmark mathematical functions in Table 2 are used to verify the performance of the ETLBO algorithm. Seen from Table 2, F_1–F_7 are uni-modal test functions used to test the convergence accuracy and solution capability of the ETLBO algorithm, F_8–F_{12} are multi-modal test functions and F_{13}–F_{20} are fixed-dimension multi-modal test functions. F_8–F_{20} are used together to test the global search capability of the ETLBO algorithm. Several state-of-the-art optimization algorithms are regarded as comparison algorithms, which are recorded in Table 3. This section compares the ETLBO algorithm with GA, ALO, DA, MFO, SCA and the original TLBO algorithms. The relevant parameters set for the seven algorithms when testing the 20 CEC benchmark test functions are shown in Table 3.

Due to the stochastic nature of meta-heuristics, the results of one single run may be unreliable. Therefore, each algorithm runs 30 times independently to reduce the statistical error. The performance of different optimization algorithms in terms of the mean and standard deviation of solutions is obtained from the 30 independent runs for 10, 30 and 50 dimensional functions. The maximal iteration 1000 is used as the stopping criterion. All experimental results are recorded in Tables 4–10. It should be noted that the closer the mean value is to the theoretical optimal value of the test function and the smaller the mean square deviation is, the better the convergence accuracy, the quality of the solution

and stability of the algorithm. In addition, the optimal performance parameters of the seven algorithms are highlighted in bold font in Tables 4–10.

Table 2. 20 benchmark test functions.

Optimization function	Value range	Optimum	Type
$F_1(x) = \sum_{i=1}^{n}(x_i)^2$	$[-100,100]^n$	0	Unimodal
$F_2(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	$[-10,10]^n$	0	Unimodal
$F_3(x) = \sum_{i=1}^{n}(\sum_{j=1}^{i} x_j)^2$	$[-100,100]^n$	0	Unimodal
$F_4(x) = \max\lvert x_i\rvert, 1 \le i \le n$	$[-100,100]^n$	0	Unimodal
$F_5(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - {x_i}^2)^2 + (x_i - 1)^2]$	$[-30,30]^n$	0	Unimodal
$F_6(x) = \sum_{i=1}^{n}(\lfloor x_i + 0.5\rfloor)^2$	$[-100,100]^n$	0	Unimodal
$F_7(x) = \sum_{i=1}^{n} i{x_i}^4 - \mathrm{random}[0,1)$	$[-1.28,1.28]^n$	0	Unimodal
$F_8(x) = \sum_{i=1}^{n}[{x_i}^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12,5.12]^n$	0	Multimodal
$F_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}{x_i}^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$[-32,32]^n$	0	Multimodal
$F_{10}(x) = \frac{1}{4000}\sum_{i=1}^{n}{x_i}^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]^n$	0	Multimodal
$F_{11}(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n}u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i+1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \le x_i < a \\ k(-x_i - a)^m, x_i < -a \end{cases}$	$[-50,50]^n$	0	Multimodal
$F_{12}(x) = 0.1\{\sin^2(3\pi x_i) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n}u(x_i, 5, 100, 4)$	$[-50,50]^n$	0	Multimodal
$F_{13}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right]^{-1}$	$[-65.536,65.536]$	1	Fixed dimension
$F_{14}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	$[-5,5]$	0.0003075	Fixed dimension
$F_{15}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5,5]$	−1.0316285	Fixed dimension
$F_{16}(x) = -\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{3}a_{ij}(x_j - p_{ij})^2\right]$	$[0,1]$	−3.86	Fixed dimension
$F_{17}(x) = -\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{6}a_{ij}(x_j - p_{ij})^2\right]$	$[0,1]$	−3.32	Fixed dimension
$F_{18}(x) = -\sum_{i=1}^{5}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0,10]$	−10.1532	Fixed dimension
$F_{19}(x) = -\sum_{i=1}^{7}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0,10]$	−10.4028	Fixed dimension
$F_{20}(x) = -\sum_{i=1}^{10}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0,10]$	−10.5363	Fixed dimension

Figure 2 shows the comparison of the convergence process curves of the seven algorithms on the 20 CEC benchmark test functions, where the horizontal and vertical axes represent the number of

iterations and the fitness value, respectively.

Table 3. Algorithm parameter settings.

Algorithm	Population size	Number of iterations	Others
GA	60	1000	pc = 0.8, pm = 0.05
ALO	60	1000	–
DA	60	1000	$\beta = 3/2$
MFO	60	1000	b = 1
SCA	60	1000	a = 2
TLBO	60	1000	–
ETLBO	60	1000	$\mu = 4$, $\eta = 40$, pm = 0.01

Table 4. Experiment comparison results on 7 uni-modal testing functions with 10 dimensions.

Function	Item	GA	ALO	DA	MFO	SCA	TLBO	ETLBO
F_1	Mean	1.01×10^{-01}	1.31×10^{-09}	5.24×10^{-01}	1.74×10^{-31}	6.91×10^{-31}	$3.73 \times 10^{+00}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	9.05×10^{-02}	5.47×10^{-10}	$1.08 \times 10^{+00}$	5.33×10^{-31}	3.09×10^{-30}	$1.72 \times 10^{+00}$	$\mathbf{0.00 \times 10^{+00}}$
F_2	Mean	4.32×10^{-02}	3.00×10^{-01}	$1.04 \times 10^{+00}$	2.07×10^{-19}	7.16×10^{-22}	$4.51 \times 10^{+00}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	2.86×10^{-02}	9.98×10^{-01}	$2.26 \times 10^{+00}$	2.90×10^{-19}	1.39×10^{-21}	$1.40 \times 10^{+00}$	$\mathbf{0.00 \times 10^{+00}}$
F_3	Mean	$1.92 \times 10^{+03}$	1.46×10^{-07}	$8.64 \times 10^{+00}$	7.21×10^{-10}	4.64×10^{-15}	$1.37 \times 10^{+01}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	$1.09 \times 10^{+03}$	2.02×10^{-07}	$1.17 \times 10^{+01}$	1.70×10^{-09}	1.49×10^{-14}	$1.15 \times 10^{+01}$	$\mathbf{0.00 \times 10^{+00}}$
F_4	Mean	$3.75 \times 10^{+00}$	3.90×10^{-05}	7.67×10^{-01}	7.03×10^{-03}	2.67×10^{-11}	9.06×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
	Std	$1.45 \times 10^{+00}$	4.26×10^{-05}	8.59×10^{-01}	3.14×10^{-02}	3.84×10^{-11}	3.56×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
F_5	Mean	$2.39 \times 10^{+02}$	$2.76 \times 10^{+01}$	$2.90 \times 10^{+02}$	$3.30 \times 10^{+01}$	$\mathbf{6.98 \times 10^{+00}}$	$2.54 \times 10^{+02}$	$8.72 \times 10^{+00}$
	Std	$5.67 \times 10^{+02}$	$8.00 \times 10^{+01}$	$4.37 \times 10^{+02}$	$8.46 \times 10^{+01}$	$\mathbf{3.07 \times 10^{-01}}$	$1.29 \times 10^{+02}$	5.27×10^{-01}
F_6	Mean	1.23×10^{-01}	1.10×10^{-09}	9.29×10^{-01}	$\mathbf{1.40 \times 10^{-31}}$	3.06×10^{-01}	$5.66 \times 10^{+00}$	6.08×10^{-01}
	Std	1.47×10^{-01}	4.26×10^{-10}	$1.43 \times 10^{+00}$	$\mathbf{2.24 \times 10^{-31}}$	1.23×10^{-01}	$2.80 \times 10^{+00}$	7.27×10^{-01}
F_7	Mean	3.07×10^{-03}	4.74×10^{-03}	7.74×10^{-03}	2.20×10^{-03}	1.20×10^{-03}	1.73×10^{-01}	2.16×10^{-03}
	Std	1.99×10^{-03}	2.96×10^{-03}	6.92×10^{-03}	1.26×10^{-03}	2.48×10^{-03}	1.36×10^{-01}	1.40×10^{-03}

The choice of 10, 30 and 50 as dimensions for benchmark functions is generally because they are representative. Lower dimensions, such as 10, can be used to evaluate the performance of algorithms on relatively smaller problem sizes. Higher dimensions, such as 50, can be used to evaluate the performance of algorithms on larger and more complex problem sizes. Additionally, selecting these specific dimensions facilitates the comparison of different algorithm performances. These dimensions have been widely used and have become standard settings for benchmark test functions. By conducting tests on these dimensions, the results become more comparable and help researchers better understand algorithm performance across different problem sizes.

All experiments were run on a 64-bit XiaoXin Air 15IKBR laptop, with an Intel(R) Core(TM) i5-

8250U CPU @ 1.60 GHz processor and 8.00 GB of RAM. The simulation software is MATLAB 2021a.

Table 5. Experiment comparison results on 7 uni-modal testing functions with 30 dimensions.

Function	Item	GA	ALO	DA	MFO	SCA	TLBO	ETLBO
F_1	Mean	$2.30 \times 10^{+03}$	1.97×10^{-07}	$2.74 \times 10^{+02}$	$3.00 \times 10^{+03}$	2.32×10^{-03}	$2.30 \times 10^{+01}$	$\mathbf{1.37 \times 10^{-21}}$
	Std	$1.59 \times 10^{+03}$	8.53×10^{-08}	$2.91 \times 10^{+02}$	$4.70 \times 10^{+03}$	4.98×10^{-03}	$6.74 \times 10^{+00}$	$\mathbf{4.97 \times 10^{-21}}$
F_2	Mean	$3.08 \times 10^{+01}$	$3.01 \times 10^{+01}$	$9.88 \times 10^{+00}$	$3.35 \times 10^{+01}$	3.33×10^{-06}	$2.28 \times 10^{+01}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	$8.55 \times 10^{+00}$	$4.35 \times 10^{+01}$	$4.28 \times 10^{+00}$	$1.57 \times 10^{+01}$	6.82×10^{-06}	$2.77 \times 10^{+00}$	$\mathbf{0.00 \times 10^{+00}}$
F_3	Mean	$3.71 \times 10^{+04}$	$1.70 \times 10^{+02}$	$5.65 \times 10^{+03}$	$1.91 \times 10^{+04}$	$1.94 \times 10^{+03}$	$1.60 \times 10^{+02}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	$1.04 \times 10^{+04}$	$1.10 \times 10^{+02}$	$7.08 \times 10^{+03}$	$1.22 \times 10^{+04}$	$2.23 \times 10^{+03}$	$7.24 \times 10^{+01}$	$\mathbf{0.00 \times 10^{+00}}$
F_4	Mean	$5.58 \times 10^{+01}$	$7.48 \times 10^{+00}$	$1.30 \times 10^{+01}$	$4.63 \times 10^{+01}$	$1.20 \times 10^{+01}$	$1.13 \times 10^{+00}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	$9.58 \times 10^{+00}$	$3.98 \times 10^{+00}$	$8.06 \times 10^{+00}$	$1.19 \times 10^{+01}$	$8.92 \times 10^{+00}$	3.05×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
F_5	Mean	$2.94 \times 10^{+05}$	$1.25 \times 10^{+02}$	$1.57 \times 10^{+04}$	$1.43 \times 10^{+04}$	$1.13 \times 10^{+02}$	$2.01 \times 10^{+03}$	$\mathbf{2.84 \times 10^{+01}}$
	Std	$4.90 \times 10^{+05}$	$2.82 \times 10^{+02}$	$3.05 \times 10^{+04}$	$3.27 \times 10^{+04}$	$2.85 \times 10^{+02}$	$8.93 \times 10^{+02}$	$\mathbf{6.16 \times 10^{-01}}$
F_6	Mean	$3.12 \times 10^{+03}$	$\mathbf{1.90 \times 10^{-07}}$	$4.23 \times 10^{+02}$	$1.00 \times 10^{+03}$	$4.13 \times 10^{+00}$	$3.18 \times 10^{+01}$	$3.39 \times 10^{+00}$
	Std	$2.40 \times 10^{+03}$	$\mathbf{1.46 \times 10^{-07}}$	$3.51 \times 10^{+02}$	$3.08 \times 10^{+03}$	3.33×10^{-01}	$6.37 \times 10^{+00}$	6.10×10^{-01}
F_7	Mean	5.61×10^{-01}	4.19×10^{-02}	8.55×10^{-02}	$1.66 \times 10^{+00}$	1.52×10^{-02}	$1.73 \times 10^{+00}$	$\mathbf{2.13 \times 10^{-03}}$
	Std	7.30×10^{-01}	1.32×10^{-02}	6.30×10^{-02}	$3.93 \times 10^{+00}$	1.32×10^{-02}	9.81×10^{-01}	$\mathbf{8.01 \times 10^{-04}}$

Table 6. Experiment comparison results on 7 uni-modal testing functions with 50 dimensions.

Function	Item	GA	ALO	DA	MFO	SCA	TLBO	ETLBO
F_1	Mean	$3.24 \times 10^{+04}$	6.24×10^{-05}	$1.19 \times 10^{+03}$	$6.50 \times 10^{+03}$	$1.68 \times 10^{+01}$	$3.32 \times 10^{+01}$	$\mathbf{2.23 \times 10^{-21}}$
	Std	$9.99 \times 10^{+03}$	3.19×10^{-05}	$9.88 \times 10^{+02}$	$9.33 \times 10^{+03}$	$2.18 \times 10^{+01}$	$9.84 \times 10^{+00}$	$\mathbf{7.03 \times 10^{-21}}$
F_2	Mean	$8.44 \times 10^{+01}$	$8.40 \times 10^{+01}$	$1.67 \times 10^{+01}$	$6.51 \times 10^{+01}$	2.55×10^{-03}	$3.78 \times 10^{+01}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	$1.83 \times 10^{+01}$	$8.38 \times 10^{+01}$	$8.36 \times 10^{+00}$	$2.66 \times 10^{+01}$	2.90×10^{-03}	$4.17 \times 10^{+00}$	$\mathbf{0.00 \times 10^{+00}}$
F_3	Mean	$1.02 \times 10^{+05}$	$3.30 \times 10^{+03}$	$1.76 \times 10^{+04}$	$4.05 \times 10^{+04}$	$2.67 \times 10^{+04}$	$3.97 \times 10^{+02}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	$2.29 \times 10^{+04}$	$1.05 \times 10^{+03}$	$1.08 \times 10^{+04}$	$1.40 \times 10^{+04}$	$1.20 \times 10^{+04}$	$1.68 \times 10^{+02}$	$\mathbf{0.00 \times 10^{+00}}$
F_4	Mean	$7.63 \times 10^{+01}$	$1.57 \times 10^{+01}$	$2.00 \times 10^{+01}$	$7.76 \times 10^{+01}$	$4.92 \times 10^{+01}$	$1.03 \times 10^{+00}$	$\mathbf{6.40 \times 10^{-12}}$
	Std	$8.07 \times 10^{+00}$	$2.30 \times 10^{+00}$	$1.11 \times 10^{+01}$	$6.55 \times 10^{+00}$	$1.08 \times 10^{+01}$	2.71×10^{-01}	$\mathbf{2.86 \times 10^{-11}}$
F_5	Mean	$2.63 \times 10^{+07}$	$2.75 \times 10^{+02}$	$1.65 \times 10^{+05}$	$4.02 \times 10^{+06}$	$2.82 \times 10^{+05}$	$3.29 \times 10^{+03}$	$\mathbf{4.83 \times 10^{+01}}$
	Std	$1.49 \times 10^{+07}$	$4.38 \times 10^{+02}$	$2.21 \times 10^{+05}$	$1.79 \times 10^{+07}$	$4.85 \times 10^{+05}$	$1.51 \times 10^{+03}$	$\mathbf{4.89 \times 10^{-01}}$
F_6	Mean	$3.33 \times 10^{+04}$	$\mathbf{6.56 \times 10^{-05}}$	$1.45 \times 10^{+03}$	$6.50 \times 10^{+03}$	$2.82 \times 10^{+01}$	$6.00 \times 10^{+01}$	$7.46 \times 10^{+00}$
	Std	$8.68 \times 10^{+03}$	$\mathbf{5.14 \times 10^{-05}}$	$9.81 \times 10^{+02}$	$8.74 \times 10^{+03}$	$3.21 \times 10^{+01}$	$1.24 \times 10^{+01}$	5.18×10^{-01}
F_7	Mean	$2.60 \times 10^{+01}$	1.22×10^{-01}	5.29×10^{-01}	$2.36 \times 10^{+01}$	2.48×10^{-01}	$6.61 \times 10^{+00}$	$\mathbf{2.21 \times 10^{-03}}$
	Std	$1.25 \times 10^{+01}$	4.61×10^{-02}	7.07×10^{-01}	$3.76 \times 10^{+01}$	1.89×10^{-01}	$3.48 \times 10^{+00}$	$\mathbf{1.08 \times 10^{-03}}$

4.1. Experiments on 7 uni-modal test functions

In this subsection, 7 uni-modal test functions are used to evaluate the performance of ETLBO. The experimental results of 10, 30 and 50 dimensional functions are listed in Tables 4, 5 and 6, respectively. From Tables 4–6, it can be seen that compared to the other six algorithms, the ETLBO algorithm can achieve smaller optimal solutions, mean values and mean square deviations for almost all uni-modal functions in 10, 30 and 50 dimensions. The results of the tests on five functions, F_1, F_2, F_3, F_4 and F_7, are particularly indicative that the ETLBO algorithm can produce satisfactory results for uni-modal functions by effectively utilizing the search space, and has good convergence accuracy and stability. Additionally, Figures 2–4 graphically present the comparison in terms of convergence speed and solution quality for solving 7 multi-modal functions (F_1, F_2, F_3, F_4, F_5, F_6 and F_7) with 10 dimensions, 30 dimensions and 50 dimensions, separately. Seen from the three figures, it is obvious that the ETLBO has the fastest convergence speed and the highest convergence accuracy on most functions compared to others.

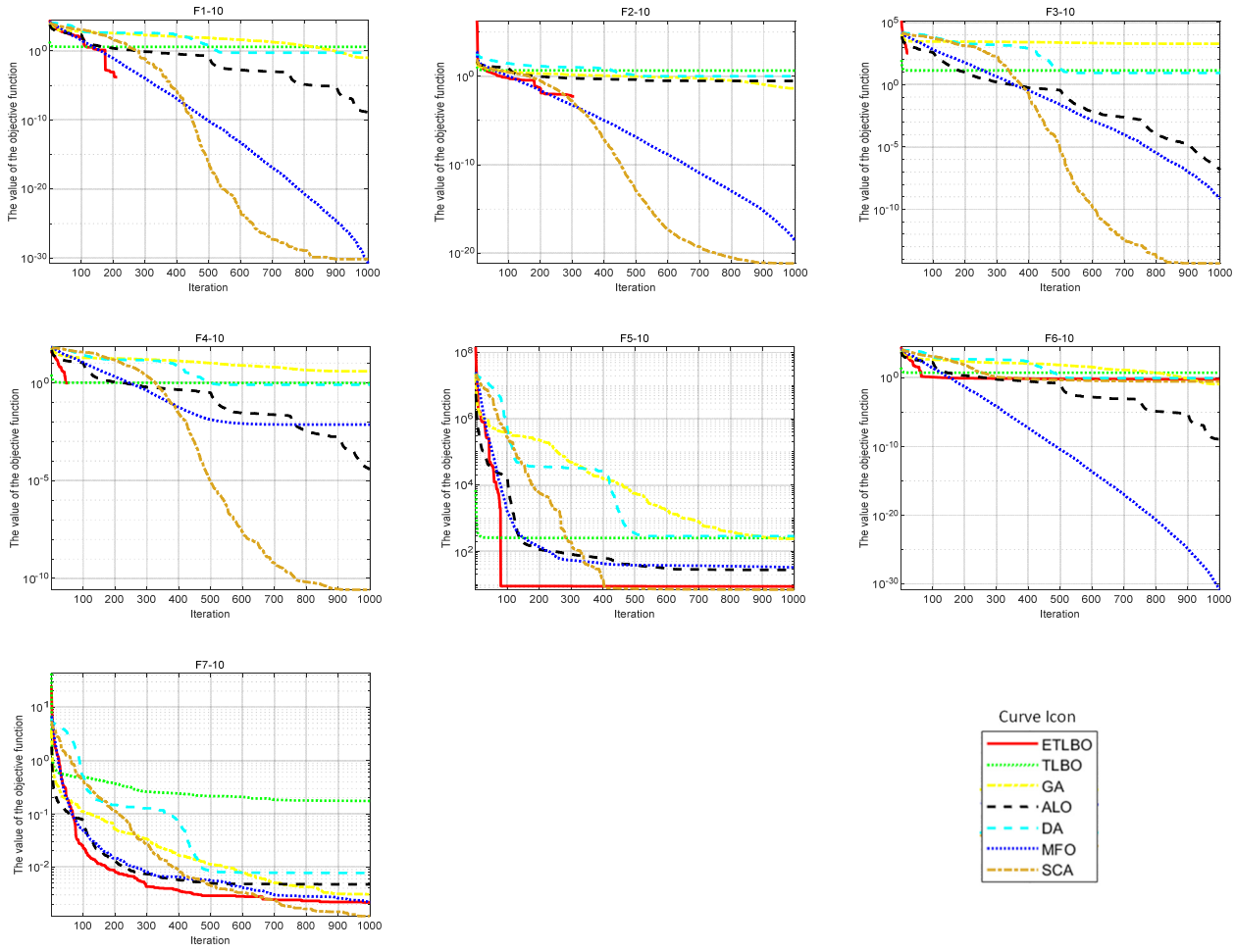

Figure 2. Simulation curves of 7 algorithms on 7 uni-modal functions with 10 dimensions.

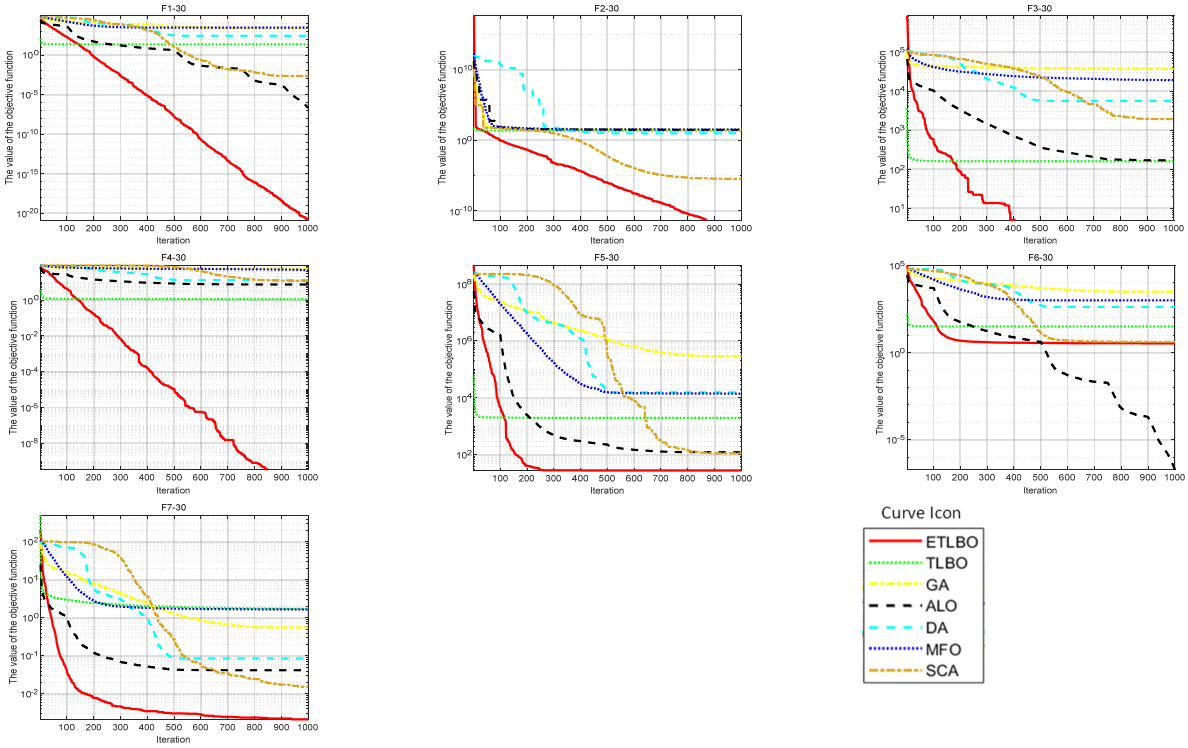

Figure 3. Simulation curves of 7 algorithms on 7 uni-modal functions with 30 dimensions.

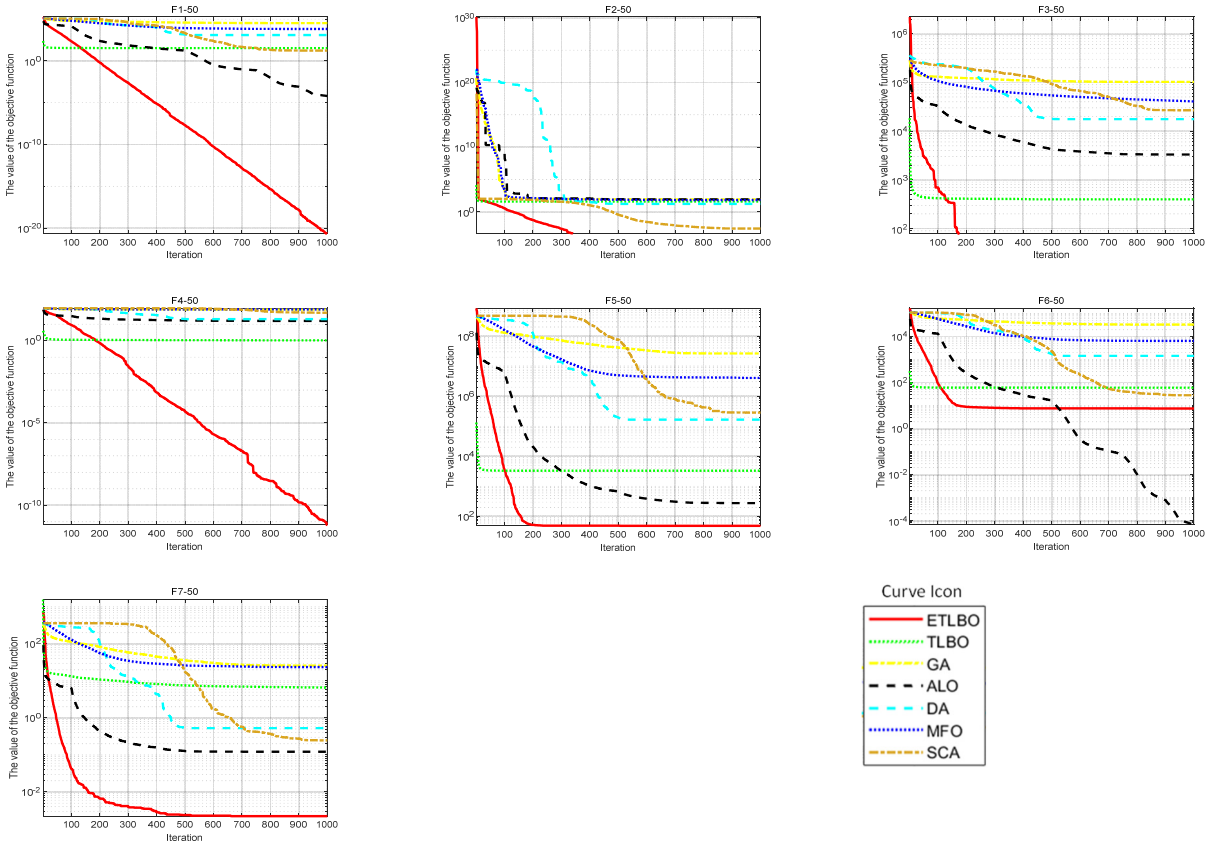

Figure 4. Simulation curves of 7 algorithms on 7 uni-modal functions with 50 dimensions.

4.2. Experiments on 5 multi-modal test functions

In this subsection, 5 multi-modal test functions are used to evaluate the performance of ETLBO. The experimental results of 10, 30 and 50 dimensional functions are listed in Tables 7, 8 and 9, respectively. Boldface in the tables indicates the best results. Good performance of an algorithm is indicated by smaller mean values. Stronger stability of an algorithm is indicated by lower standard deviation values. According to these tables, the proposed ETLBO algorithm presents superior performance on most functions. Seen from Table 7, the performance of ETLBO is better than other algorithms for functions F_8, F_9, F_{10} and F_{11}. The ALO has the smallest mean and standard deviation for function F_{12}. Seen from Tables 8 and 9, the ETLBO still shows the best performance on functions F_8, F_9, F_{10} and F_{11} compared to other algorithms. In brief, the proposed ETLBO improves the solution quality for multi-modal functions. Additionally, Figures 5–7 graphically present the comparison in terms of convergence speed and solution quality for solving 5 multi-modal functions (F_8, F_9, F_{10}, F_{11} and F_{12}) with 10 dimensions, 30 dimensions and 50 dimensions, separately. Seen from the three figures, it is obvious that ETLBO has the fastest convergence speed and the highest convergence accuracy on most functions compared to other algorithms.

Table 7. Experiment comparison results on 5 multi-modal testing functions with 10 dimensions.

Function	Item	GA	ALO	DA	MFO	SCA	TLBO	ETLBO
F_8	Mean	$3.11 \times 10^{+00}$	$1.73 \times 10^{+01}$	$1.74 \times 10^{+01}$	$1.58 \times 10^{+01}$	$1.19 \times 10^{+00}$	$3.71 \times 10^{+01}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	$2.23 \times 10^{+00}$	$9.42 \times 10^{+00}$	$1.30 \times 10^{+01}$	$1.03 \times 10^{+01}$	$5.31 \times 10^{+00}$	$8.29 \times 10^{+00}$	$\mathbf{0.00 \times 10^{+00}}$
F_9	Mean	$1.50 \times 10^{+00}$	1.72×10^{-05}	$1.56 \times 10^{+00}$	4.00×10^{-15}	3.29×10^{-15}	$3.59 \times 10^{+00}$	$\mathbf{4.44 \times 10^{-16}}$
	Std	$3.09 \times 10^{+00}$	3.91×10^{-06}	$1.26 \times 10^{+00}$	$0.00 \times 10^{+00}$	1.46×10^{-15}	6.80×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
F_{10}	Mean	6.13×10^{-01}	2.09×10^{-01}	2.49×10^{-01}	1.66×10^{-01}	2.71×10^{-03}	4.19×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
	Std	3.43×10^{-01}	1.27×10^{-01}	2.43×10^{-01}	1.05×10^{-01}	8.43×10^{-03}	1.53×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
F_{11}	Mean	9.34×10^{-02}	$1.14 \times 10^{+00}$	5.46×10^{-01}	7.79×10^{-02}	6.04×10^{-02}	9.32×10^{-01}	$\mathbf{3.25 \times 10^{-02}}$
	Std	2.82×10^{-01}	$2.62 \times 10^{+00}$	6.22×10^{-01}	3.48×10^{-01}	2.56×10^{-02}	4.57×10^{-01}	7.27×10^{-02}
F_{12}	Mean	1.42×10^{-02}	$\mathbf{1.10 \times 10^{-03}}$	3.17×10^{-01}	4.39×10^{-03}	2.18×10^{-01}	2.97×10^{-01}	7.43×10^{-01}
	Std	1.85×10^{-02}	$\mathbf{3.38 \times 10^{-03}}$	3.54×10^{-01}	5.52×10^{-03}	7.16×10^{-02}	2.22×10^{-01}	3.44×10^{-01}

Table 8. Experiment comparison results on 5 multi-modal testing functions with 30 dimensions.

Function	Item	GA	ALO	DA	MFO	SCA	TLBO	ETLBO
F_8	Mean	$1.13 \times 10^{+02}$	$7.08 \times 10^{+01}$	$7.87 \times 10^{+01}$	$1.44 \times 10^{+02}$	$1.37 \times 10^{+01}$	$1.85 \times 10^{+02}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	$3.81 \times 10^{+01}$	$2.01 \times 10^{+01}$	$4.55 \times 10^{+01}$	$2.50 \times 10^{+01}$	$2.39 \times 10^{+01}$	$2.32 \times 10^{+01}$	$\mathbf{0.00 \times 10^{+00}}$
F_9	Mean	$1.89 \times 10^{+01}$	$1.93 \times 10^{+00}$	$4.99 \times 10^{+00}$	$1.54 \times 10^{+01}$	$1.14 \times 10^{+01}$	$4.48 \times 10^{+00}$	$\mathbf{4.44 \times 10^{-16}}$
	Std	6.46×10^{-01}	4.96×10^{-01}	$1.18 \times 10^{+00}$	$8.03 \times 10^{+00}$	$9.72 \times 10^{+00}$	3.32×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
F_{10}	Mean	$2.71 \times 10^{+01}$	9.10×10^{-03}	$3.11 \times 10^{+00}$	$1.81 \times 10^{+01}$	8.34×10^{-02}	7.49×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
	Std	$2.40 \times 10^{+01}$	8.14×10^{-03}	$2.07 \times 10^{+00}$	$3.71 \times 10^{+01}$	1.39×10^{-01}	1.18×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
F_{11}	Mean	$1.03 \times 10^{+01}$	$7.59 \times 10^{+00}$	$4.98 \times 10^{+00}$	4.20×10^{-01}	$1.96 \times 10^{+01}$	$1.50 \times 10^{+00}$	$\mathbf{2.54 \times 10^{-01}}$
	Std	$4.55 \times 10^{+00}$	$3.45 \times 10^{+00}$	$4.26 \times 10^{+00}$	7.31×10^{-01}	$8.44 \times 10^{+01}$	7.26×10^{-01}	$\mathbf{3.10 \times 10^{-01}}$
F_{12}	Mean	$3.91 \times 10^{+04}$	$\mathbf{7.87 \times 10^{-03}}$	$2.60 \times 10^{+02}$	2.17×10^{-01}	$6.64 \times 10^{+00}$	$1.28 \times 10^{+00}$	$2.16 \times 10^{+00}$
	Std	$6.87 \times 10^{+04}$	$\mathbf{1.20 \times 10^{-02}}$	$5.47 \times 10^{+02}$	6.43×10^{-01}	$1.72 \times 10^{+01}$	2.29×10^{-01}	4.94×10^{-01}

Table 9. Experiment comparison results on 5 multi-modal testing functions with 50 dimensions.

Function	Item	GA	ALO	DA	MFO	SCA	TLBO	ETLBO
F_8	Mean	$3.97 \times 10^{+02}$	$1.30 \times 10^{+02}$	$1.25 \times 10^{+02}$	$2.86 \times 10^{+02}$	$6.09 \times 10^{+01}$	$3.44 \times 10^{+02}$	$\mathbf{0.00 \times 10^{+00}}$
	Std	$5.88 \times 10^{+01}$	$3.16 \times 10^{+01}$	$7.61 \times 10^{+01}$	$6.52 \times 10^{+01}$	$5.34 \times 10^{+01}$	$2.81 \times 10^{+01}$	$\mathbf{0.00 \times 10^{+00}}$
F_9	Mean	$1.99 \times 10^{+01}$	$2.89 \times 10^{+00}$	$6.81 \times 10^{+00}$	$1.85 \times 10^{+01}$	$1.52 \times 10^{+01}$	$4.55 \times 10^{+00}$	$\mathbf{4.44 \times 10^{-16}}$
	Std	5.20×10^{-01}	$1.18 \times 10^{+00}$	$2.08 \times 10^{+00}$	$3.00 \times 10^{+00}$	$8.48 \times 10^{+00}$	3.74×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
F_{10}	Mean	$3.15 \times 10^{+02}$	1.23×10^{-02}	$2.23 \times 10^{+01}$	$6.83 \times 10^{+01}$	$1.29 \times 10^{+00}$	7.39×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
	Std	$8.03 \times 10^{+01}$	5.79×10^{-03}	$2.39 \times 10^{+01}$	$1.01 \times 10^{+02}$	7.63×10^{-01}	1.22×10^{-01}	$\mathbf{0.00 \times 10^{+00}}$
F_{11}	Mean	$6.24 \times 10^{+06}$	$1.20 \times 10^{+01}$	$9.64 \times 10^{+01}$	$6.40 \times 10^{+07}$	$1.94 \times 10^{+05}$	$1.48 \times 10^{+00}$	$\mathbf{3.77 \times 10^{-01}}$
	Std	$6.29 \times 10^{+06}$	$4.26 \times 10^{+00}$	$2.85 \times 10^{+02}$	$1.41 \times 10^{+08}$	$3.57 \times 10^{+05}$	7.12×10^{-01}	$\mathbf{5.12 \times 10^{-02}}$
F_{12}	Mean	$5.00 \times 10^{+07}$	$4.27 \times 10^{+01}$	$6.54 \times 10^{+04}$	$1.60 \times 10^{+03}$	$1.34 \times 10^{+06}$	$\mathbf{1.91 \times 10^{+00}}$	$4.61 \times 10^{+00}$
	Std	$3.22 \times 10^{+07}$	$3.51 \times 10^{+01}$	$1.33 \times 10^{+05}$	$6.73 \times 10^{+03}$	$4.10 \times 10^{+06}$	4.58×10^{-01}	$\mathbf{3.54 \times 10^{-01}}$

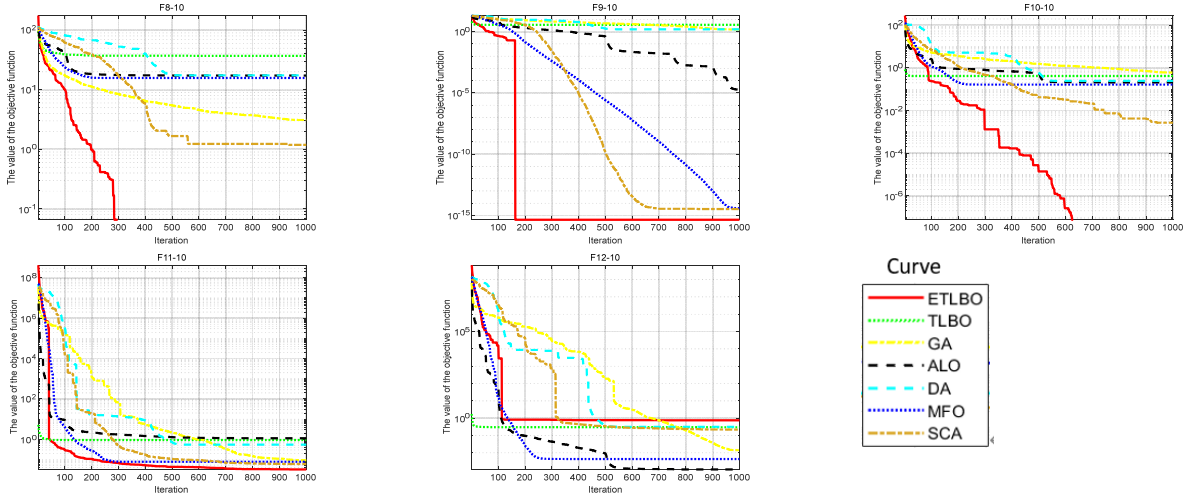

Figure 5. Simulation curves of 7 algorithms on 5 multi-modal functions with 10 dimensions.

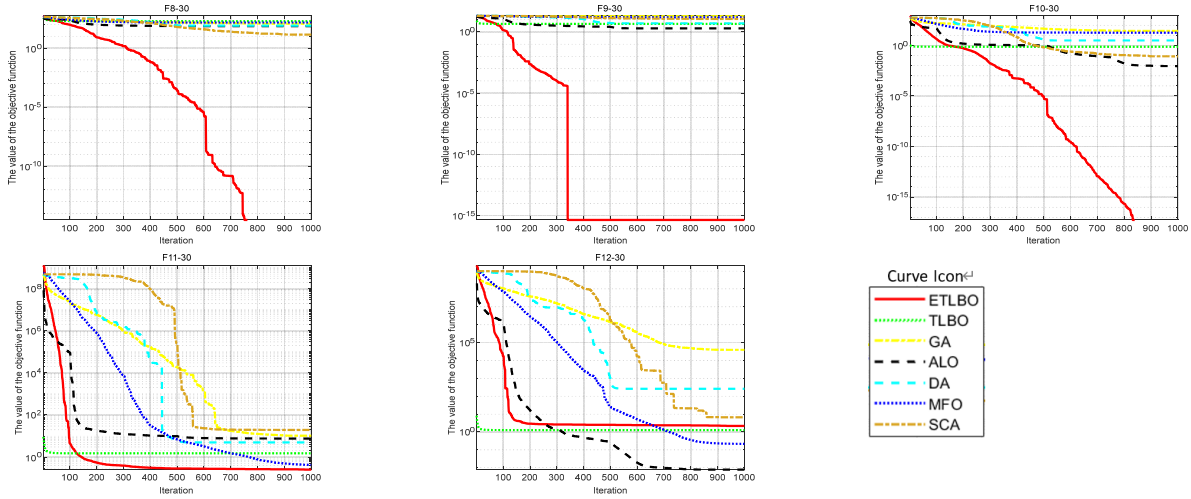

Figure 6. Simulation curves of 7 algorithms on 5 multi-modal functions with 30 dimensions.

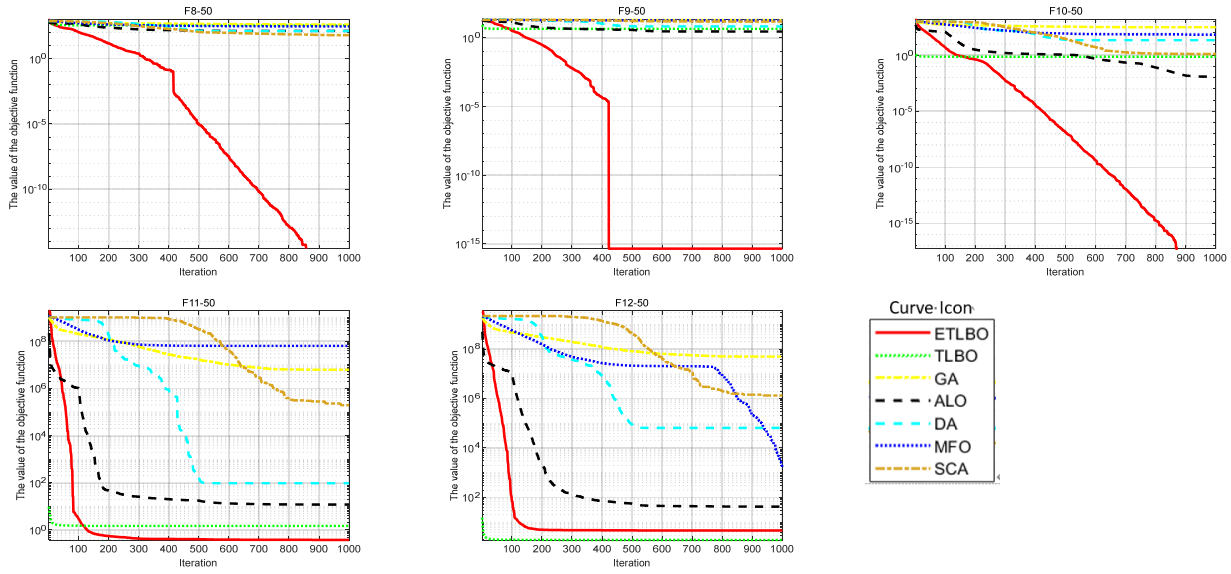

Figure 7. Simulation curves of 7 algorithms on 5 multi-modal functions with 50 dimensions.

4.3. Experiments on 8 fixed dimension test functions

In this subsection, 8 fixed dimension functions are applied to evaluate the performance of the proposed ETLBO algorithm. The experimental results are listed in Table 10. According to this table, the ETLBO algorithm presents the best solution on five functions (F_{15}, F_{16}, F_{18}, F_{19}, F_{20}). Therefore, the ETLBO is the most fit to address multi-modal testing functions with fixed dimensions.

Table 10. Experiment comparison results on 8 multi-modal testing functions with fixed dimension.

Function	Item	GA	ALO	DA	MFO	SCA	TLBO	ETLBO
F_{13}	Mean	**$1.05 \times 10^{+00}$**	$1.15 \times 10^{+00}$	$1.05 \times 10^{+00}$	$1.29 \times 10^{+00}$	$1.20 \times 10^{+00}$	$1.27 \times 10^{+01}$	$4.49 \times 10^{+00}$
	Std	2.22×10^{-01}	3.64×10^{-01}	2.22×10^{-01}	$1.11 \times 10^{+00}$	6.11×10^{-01}	4.29×10^{-09}	$5.08 \times 10^{+00}$
F_{14}	Mean	5.10×10^{-03}	1.74×10^{-03}	1.47×10^{-03}	9.39×10^{-04}	**8.67×10^{-04}**	1.13×10^{-03}	3.70×10^{-03}
	Std	6.34×10^{-03}	4.39×10^{-03}	4.72×10^{-04}	3.09×10^{-04}	3.41×10^{-04}	4.64×10^{-04}	9.02×10^{-03}
F_{15}	Mean	$-1.00 \times 10^{+00}$	**$-1.03 \times 10^{+00}$**	**$-1.03 \times 10^{+00}$**	**$-1.03 \times 10^{+00}$**	**$-1.03 \times 10^{+00}$**	**$-1.03 \times 10^{+00}$**	**$-1.03 \times 10^{+00}$**
	Std	5.11×10^{-02}	2.00×10^{-14}	1.22×10^{-08}	2.28×10^{-16}	8.50×10^{-06}	6.19×10^{-03}	**5.69×10^{-07}**
F_{16}	Mean	$-3.44 \times 10^{+00}$	$-3.86 \times 10^{+00}$	$-3.86 \times 10^{+00}$	$-3.86 \times 10^{+00}$	$-3.86 \times 10^{+00}$	$-3.81 \times 10^{+00}$	**$-3.86 \times 10^{+00}$**
	Std	2.41×10^{-01}	1.26×10^{-05}	1.56×10^{-03}	9.31×10^{-15}	3.06×10^{-03}	7.23×10^{-02}	**2.28×10^{-15}**
F_{17}	Mean	$-1.73 \times 10^{+00}$	**$-3.26 \times 10^{+00}$**	**$-3.26 \times 10^{+00}$**	$-3.21 \times 10^{+00}$	$-3.04 \times 10^{+00}$	$-2.76 \times 10^{+00}$	$-3.24 \times 10^{+00}$
	Std	4.81×10^{-01}	6.03×10^{-02}	7.17×10^{-02}	5.61×10^{-02}	2.18×10^{-01}	3.88×10^{-01}	**4.54×10^{-02}**
F_{18}	Mean	$-1.49 \times 10^{+00}$	$-7.88 \times 10^{+00}$	$-9.39 \times 10^{+00}$	$-5.64 \times 10^{+00}$	$-3.00 \times 10^{+00}$	$-8.80 \times 10^{+00}$	**$-1.02 \times 10^{+01}$**
	Std	$1.06 \times 10^{+00}$	$2.94 \times 10^{+00}$	$1.85 \times 10^{+00}$	$3.20 \times 10^{+00}$	$2.00 \times 10^{+00}$	$2.41 \times 10^{+00}$	**1.37×10^{-04}**
F_{19}	Mean	$-1.59 \times 10^{+00}$	$-6.62 \times 10^{+00}$	$-1.04 \times 10^{+01}$	$-9.42 \times 10^{+00}$	$-4.85 \times 10^{+00}$	$-8.28 \times 10^{+00}$	**$-1.04 \times 10^{+01}$**
	Std	6.67×10^{-01}	$3.29 \times 10^{+00}$	5.29×10^{-03}	$2.43 \times 10^{+00}$	$1.93 \times 10^{+00}$	$2.44 \times 10^{+00}$	**5.51×10^{-05}**
F_{20}	Mean	$-1.58 \times 10^{+00}$	$-7.92 \times 10^{+00}$	$-1.02 \times 10^{+01}$	$-9.77 \times 10^{+00}$	$-5.43 \times 10^{+00}$	$-9.63 \times 10^{+00}$	**$-1.05 \times 10^{+01}$**
	Std	5.22×10^{-01}	$3.37 \times 10^{+00}$	$1.20 \times 10^{+00}$	$2.37 \times 10^{+00}$	$1.11 \times 10^{+00}$	$2.22 \times 10^{+00}$	**5.86×10^{-05}**

5 Boiler combustion optimization

In this section, the proposed ETLBO algorithm is used to optimize the boiler's adjustment parameters for reducing the NO_x emission concentration. First, based on the boiler operation parameters, extreme learning machine (ELM) [52] is applied to build NO_x emission models. Secondly, based on the building model, the ETLBO is used to reduce NO_x emission. Note that the extreme learning machine is an effective modeling method that has been applied in many fields. Therefore, detailed description of extreme learning machine can be found in Reference [52], whereas this paper only gives the applicable rules of extreme learning machine.

5.1. Analysis and modeling of NO_x emissions

This section focuses on using ELM to establish a prediction model of NO_x emissions. For a 330 MW circulating fluidized bed boiler (CFBB), a total of 28,800 sets of operation data were collected, including 26 input parameters and one output parameter. A detailed description of these data can be found in reference [53]. Considering the small data sampling interval and low data fluctuation in the CFBB, this paper sets the data sampling interval as every 80 units, resulting in a total of 360 datasets to fully ensure the generalization ability of the prediction model, which is divided into three parts: training data, validation data and testing data. The proportions are 65, 15 and 20%, respectively. The training data are used to establish the prediction model and determine the model parameters. The validation data are used to verify the effectiveness of the prediction model. The test data are used to test the generalization ability of the prediction model. In this paper, the ELM algorithm model is configured with 26 input nodes, 41 hidden layer neural nodes and 1 output node, and the "Sigmoid" function is set as the hidden layer activation function. Note that in order to make these boilers' operation data dimension unity, they are processed by min-max normalization method.

In order to prove the excellent accuracy and effectiveness of the NO_x emission prediction model established using extreme learning machine, this section conducted 30 independent test experiments and obtained the corresponding average results, which are recorded in Table 11. Note that experiment results are obtained after normalization in Table 11. From the testing results, it can be observed that the mean value reaches the precision of 10^{-2}, indicating that the prediction model has a high level of accuracy. The S.D. value reaches the precision of 10^{-3}, indicating that the prediction model exhibits good stability. The R^2 value is extremely close to 1, demonstrating the prediction model has strong generalization and regression capabilities. In addition, the mean absolute percentage error (MAPE) value reaches the precision of 10^{-4} or 10^{-5} and the mean absolute error (MAE) value reaches the precision of 10^{-2}, proving that the prediction model has the ability to approximate the target values effectively.

In Figures 8 and 9, the solid line with red asterisks represents the actual NO_x emission of one boiler, while the dotted line with black circles represents the predicted NO_x emission of the ELM model. Seen from Figures 8 and 9, the predicted NO_x emission can almost match the actual NO_x emission. Therefore, the NO_x emission model built by ELM is effective.

Table 11. Performance index statistics for NO_x emission model.

Performance index	Training sample	Validation sample	Testing sample
Mean	0.0813	0.0969	0.0959
S.D.	0.0033	0.0069	0.0065
MAPE	7.01×10^{-05}	9.12×10^{-04}	9.88×10^{-04}
MAE	0.0633	0.0761	0.0756
R^2	0.8944	0.8356	0.8241

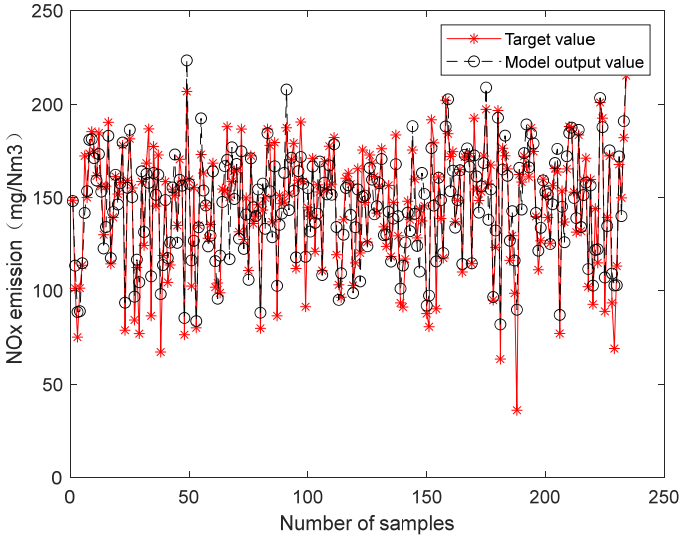

Figure 8. NO_x emission training model curve.

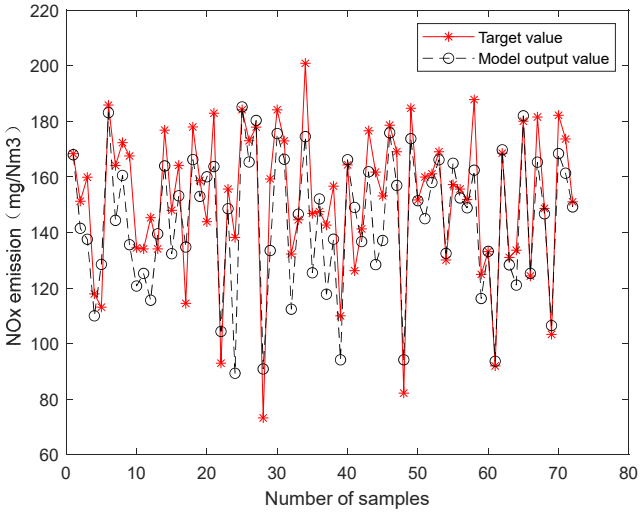

Figure 9. NO_x emission testing model curve.

5.2. Optimizing NO_x emissions

In this subsection, based on the established NO_x emission prediction model by ELM, the ETLBO algorithm is applied to optimize the adjustment parameters of the CFBB for reducing NO_x emissions.

In the process of optimizing NO_x emissions, the main focus is on optimizing 11 adjustable parameters that have a significant impact on NO_x emissions, while keeping the remaining parameters unchanged. The specific details are shown in Table 12. The objective function for optimizing NO_x emissions is as follows:

$$\min\ f(x) = f_{NO_x}(x), \tag{16}$$

$$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}], \tag{17}$$

$$s.t.\, a_i \le x_i \le b_i. \tag{18}$$

Table 12. The 11 key parameters that affect NO_x emissions.

Variable	Variable meaning	Variable	Variable meaning
x	Parameters to be optimized	x_1	Coal feed amount A
x_2	Coal feed amount B	x_3	Coal feed amount C
x_4	Coal feed amount D	x_5	Primary air flow rate at the entrance of the left-side air duct burner
x_6	Primary air flow rate at the entrance of the right-side air duct burner	x_7	Total flow rate of secondary air on the left side
x_8	Total flow rate of secondary air on the right side	x_9	Distribution flow rate of internal secondary air on the left side
x_{10}	Distribution flow rate of internal secondary air on the right side	x_{11}	Flue gas oxygen concentration
a_i	The lower limit value of the i-th parameter to be optimized	b_i	The upper limit value of the i-th parameter to be optimized

Based on the established NO_x emission prediction model, the ETLBO algorithm is applied to tune the adjustable parameters in order to achieve the goal of reducing NO_x emissions. In this section, the maximum number of iterations for the ETLBO algorithm is set to 50, the population size is set to 40 and the dimension is set to 11. Other parameters remain unchanged. The operating condition data is then optimized. Figure 10 shows the comparison curve before and after the optimization of NO_x emissions, where the solid line with red asterisks represents the data before optimization, and the dotted line with blue circles represents the data after optimization. Seen from Figure 10, it can be visually observed that after optimizing 72 sets of data using the ETLBO algorithm, there is a certain degree of reduction in NO_x emissions. This proves that the NO_x emission model based on the ELM algorithm is effective, and the ETLBO algorithm proposed in this paper is an effective strategy for solving complex global optimization problems.

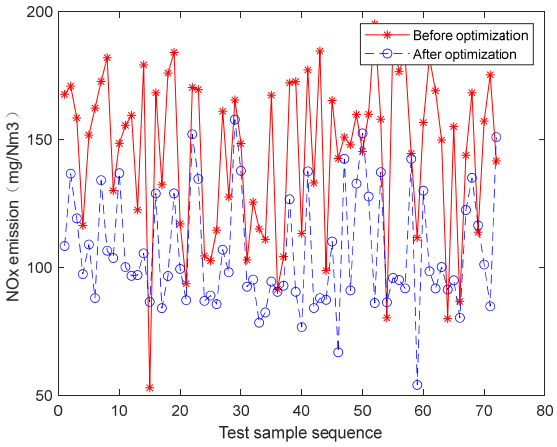

Figure 10. The comparison curve before and after the optimization of NO_x emissions

According to Table 13, in this section of the experiment, three sets of data, D, E and F, were randomly selected from the test data. By comparing the data before and after parameter optimization, it can be visually observed that the NO_x emissions were significantly reduced. The coal feed amount is reduced, the primary air flow rate increased and the flue gas oxygen concentration decreased after optimization for samples D, E and F. The secondary air flow rate of sample D decreased, but the secondary air flow rate of samples E and F both increased. In the end, the NO_x emissions for samples D, E and F are reduced by 93.8604 mg/Nm^3, 75.5935 mg/Nm^3 and 27.0340 mg/Nm^3, respectively. Therefore, considering only the reduction of NO_x emissions, the ETLBO-ELM method proposed in this paper is an effective strategy.

Table 13. Comparison of parameters before and after optimization of NO_x emissions

Boiler adjustable parameters		Test sample data D		Test sample data E		Test sample data F	
		Before	After	Before	After	Before	After
Coal feed amount	A	56.065	54.646	52.997	44.653	49.900	48.321
	B	54.653	53.258	44.670	40.369	49.536	48.345
	C	55.083	54.080	44.581	40.656	46.110	44.685
	D	55.434	54.431	52.812	44.089	43.499	42.101
Primary air flow rate	left	260.452	299.130	223.604	239.854	171.193	228.639
	right	210.330	301.190	251.297	316.067	212.389	299.588
Total flow rate of secondary air	left	451.981	433.481	353.973	553.023	155.050	373.982
Total flow rate of internal secondary air	left	613.652	598.490	622.234	785.970	292.283	355.985
Total flow rate of secondary air	right	1135.310	1007.080	1001.858	896.991	721.062	612.566
Total flow rate of internal secondary air	right	718.550	733.712	621.376	629.101	318.889	320.510
Flue gas oxygen concentration		4.799	3.179	5.371	3.689	5.327	5.018
NO_x emission (mg/Nm^3)		189.7310	95.8706	182.1030	106.5095	159.8260	132.7920

6. Conclusions

To enhance the performance of the original TLBO algorithm, an evolution TLBO algorithm is proposed. Compared to TLBO, the proposed ETLBO uses the chaotic function to initialize population individuals, introduces the inertia weight, acceleration coefficient and self-adaptive teaching factors into the teaching phase, and the idea of heredity is used to update the population in the learning phase. 20 benchmark test functions are used to verify the performance of ETLBO and experimental results show that the ETLBO outperforms the conventional TLBO on most test functions. Therefore, the ETLBO has good convergence ability. Additionally, the ETLBO combines with extreme learning machine to solve the boiler combustion optimization problem. Experimental results reveal that NO_x emissions can be reduced. In conclusion, the ETLBO algorithm is an effective optimization method.

In the future, the performance of the ETLBO algorithm will be further improved and applied to engineering optimization problems. Additionally, further research is needed to provide rigorous mathematical proofs for the convergence of the ETLBO algorithm. The multi-objective version of the ETLBO algorithm and its application in uncertain engineering problems also deserve further study.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

Funding: This work is supported the National Natural Science Foundation of China (Grant No. 62203332), the Natural Science Foundation of Tianjin (Grant No.20JCQNJC00430) and College Students' Innovative Entrepreneurial Training Plan Program (Grant No. 202310069032).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. F. Zou, L. Wang, X. Hei, D. Chen, Teaching-learning-based optimization with learning experience of other learners and its application, *Appl. Soft Comput.*, **37** (2015), 725–736. https://doi.org/10.1016/j.asoc.2015.08.047
2. S. Yu, S. Su, Research and application of chaotic glowworm swarm optimization algorithm, *J. Front. Comput. Sci. Technol.*, **8** (2014), 352–358. https://doi.org/10.3778/j.issn.1673-9418.1310016
3. S. He, Q. H. Wu, J. Saunders, Group search optimizer: An optimization algorithm inspired by animal searching behavior, *IEEE Trans. Evolut. Comput.*, **13** (2009), 973–990. https://doi.org/10.1109/TEVC.2009.2011992
4. D. Karaboga, B. Akay, A comparative study of Artificial Bee Colony algorithm, *Appl. Math. Comput.*, **214** (2009), 108–132. https://doi.org/10.1016/j.amc.2009.03.090

5. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95-International Conference on Neural Networks*, **4** (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
6. M. Dorigo, V. Maniezzo, A. Colorni, Ant System: Optimization by a colony of cooperating agents, *IEEE Trans. Syst., Man, Cybern*., *Part B*, **26** (1996), 29–41. https://doi.org/10.1109/3477.484436
7. M. M. Eusuff, K. E. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, *J. Water Resour. Plann. Manage.*, **129** (2003), 210–225. https://doi.org/10.1061/(ASCE)0733-9496(2003)129:3(210)
8. D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *J. Global Optim.*, **39** (2007), 459–471. https://doi.org/10.1007/s10898-007-9149-x
9. X. Li, Z. Shao, J. Qian, An optimizing method based on autonomous animats: Fish swarm algorithm, *Syst. Eng.-Theory Pract.*, **11** (2002), 32–38.
10. S. Mirjalili, S. Saremi, S. M. Mirjalili, L. S. Coelho, Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization, *Expert Syst. Appl.*, **47** (2016), 106–119. https://doi.org/10.1016/j.eswa.2015.10.039
11. K. M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control Syst. Mag.*, **22** (2002), 52–67. https://doi.org/10.1109/MCS.2002.1004010
12. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
13. F. A. Hashim, A. G. Hussien, Snake Optimizer: A novel meta-heuristic optimization algorithm, *Knowl.-Based Syst.*, **242** (2022), 108320. https://doi.org/10.1016/j.knosys.2022.108320
14. E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: A Gravitational Search Algorithm, *Inf. Sci.*, **179** (2009), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
15. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, Boston, 1989.
16. S. Mirjalili, The ant lion optimizer, *Adv. Eng. Software*, **83** (2015), 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010
17. S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Comput. Appl.*, **27** (2016), 1053–1073. https://doi.org/10.1007/s00521-015-1920-1
18. S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.*, **89** (2015), 228–249. https://doi.org/10.1016/j.knosys.2015.07.006
19. S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, *Knowl.-Based Syst.*, **96** (2016), 120–133. https://doi.org/10.1016/j.knosys.2015.12.022
20. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, *Comput.-Aided Des.*, **43** (2011), 303–315. https://doi.org/10.1016/j.cad.2010.12.015
21. R. V. Rao, V. Patel, An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems, *Sci. Iran.*, **20** (2013), 710–720. https://doi.org/10.1016/j.scient.2012.12.005
22. K. Yu, X. Wang, Z. Wang, Elitist teaching-learning-based optimization algorithm based on feedback, *Acta Autom. Sin.*, **40** (2014), 1976–1983.

23. L. Gao, H. Ouyang, X. Kong, H. Liu, Teaching-learning based optimization algorithm with crossover operation, *J. Northeastern Univ. (Nat. Sci.)*, **35** (2014), 323–327. https://doi.org/10.3969/j.issn.1005-3026.2014.03.005
24. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-Learning-Based Optimization: An optimization method for continuous non-linear large scale problems, *Inf. Sci.*, **183** (2012), 1–15. https://doi.org/10.1016/j.ins.2011.08.006
25. S. Yang, Y. Zhang, S. Xu, Z. Liao, J. Li, Parameter identification of photovoltaic cell model based on grouping teaching-learning-based optimization algorithm, *Distrib. Energy*, **7** (2022), 52–61. https://doi.org/10.16513/j.2096-2185.DE.2207307
26. T. Niknam, R. Azizipanah-Abarghooee, M. Rasoul Narimani, A new multi objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems, *Eng. Appl. Artif. Intell.*, **25** (2012), 1577–1588. https://doi.org/10.1016/j.engappai.2012.07.004
27. S. C. Satapathy, A. Naik, Data clustering based on teaching-learning-based optimization, in *International Conference on Swarm, Evolutionary, and Memetic Computing*, **7077** (2011), 148–156. https://doi.org/10.1007/978-3-642-27242-4_18
28. A. B. Gunji, B. B. B. V. L. Deepak, C. M. V. A. R. Bahubalendruni, D. B. B. Biswal, An optimal robotic assembly sequence planning by assembly subsets detection method using teaching-learning-based optimization algorithm, *IEEE Trans. Autom. Sci. Eng.*, **15** (2018), 1369–1385. https://doi.org/10.1109/TASE.2018.2791665
29. C. Wu, Y. He, J. Zhao, Solving set-union knapsack problem by modified teaching-learning-based optimization algorithm, *J. Front. Comput. Sci. Technol.*, **12** (2018), 2007–2020. https://doi.org/10.3778/j.issn.1673-9418.1802021
30. M. Ghasemi, S. Ghavidel, M. Gitizadeh, E. Akbari, An improved teaching-learning-based optimization algorithm using Lévy mutation strategy for non-smooth optimal power flow, *Int. J. Electr. Power Energy Syst.*, **65** (2015), 375–384. https://doi.org/10.1016/j.ijepes.2014.10.027
31. G. Li, P. Niu, W. Zhang, Y. Liu, Model NOx emissions by least squares support vector machine with tuning based on ameliorated teaching-learning-based optimization, *Chemom. Intell. Lab. Syst.*, **126** (2013), 11–20. https://doi.org/10.1016/j.chemolab.2013.04.012
32. B. Wang, H. Li, Y. Feng, An improved teaching-learning-based optimization for constrained evolutionary optimization, *Inf. Sci.*, **456** (2018), 131–144. https://doi.org/10.1016/j.ins.2018.04.083
33. K. Yu, X. Wang, Z. Wang, An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems, *J. Intell. Manuf.*, **27** (2016), 831–843. https://doi.org/10.1007/s10845-014-0918-3
34. H. Tsai, Confined teaching learning based optimization with variable search strategies for continuous optimization, *Inf. Sci.*, **500** (2019), 34–47. https://doi.org/10.1016/j.ins.2019.05.065
35. R. V. Rao, V. Patel, An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems, *Int. J. Ind. Eng. Comput.*, **3** (2012), 535–560. https://doi.org/10.5267/j.ijiec.2012.03.007
36. F. Zou, L. Wang, X. Hei, D. Chen, D. Yang, Teaching-learning-based optimization with dynamic group strategy for global optimization, *Inf. Sci.*, **273** (2014), 112–131. https://doi.org/10.1016/j.ins.2014.03.038

37. D. Chen, R. Lu, F. Zou, S. Li, Teaching-learning-based optimization with variable-population scheme and its application for ANN and global optimization, *Neurocomputing*, **173** (2016), 1096–1111. https://doi.org/10.1016/j.neucom.2015.08.068
38. S. Sultana, P. K. Roy, Multi-objective quasi-oppositional teaching learning based optimization for optimal location of distributed generator in radial distribution systems, *Int. J. Electr. Power Energy Syst.*, **63** (2014), 534–545. https://doi.org/10.1016/j.ijepes.2014.06.031
39. F. Zou, D. Chen, Q. Xu, A survey of teaching-learning-based optimization, *Neurocomputing*, **335** (2019), 366–383. https://doi.org/10.1016/j.neucom.2018.06.076
40. S. Tuo, L. Yong, F. Deng, Y. Li, Y. Lin, Q. Lu, HSTLBO: A hybrid algorithm based on Harmony Search and Teaching-Learning-Based Optimization for complex high-dimensional optimization problems, *Plos One*, **12** (2017), 0175114. https://doi.org/10.1371/journal.pone.0175114
41. X. Li, P. Niu, J. Liu, Combustion optimization of a boiler based on the chaos and Lévy flight vortex search algorithm, *Appl. Math. Modell.*, **58** (2018), 3–18. https://doi.org/10.1016/j.apm.2018.01.043
42. Y. Niu, J. Kang, F. Li, W. Ge, G. Zhou, Case-based reasoning based on grey-relational theory for the optimization of boiler combustion systems, *ISA Trans.*, **103** (2020), 166–176. https://doi.org/10.1016/j.isatra.2020.03.024
43. Y. Shi, W. Zhong, X. Chen, A. B. Yu, Jie Li, Combustion optimization of ultra supercritical boiler based on artificial intelligence, *Energy*, **170** (2019), 804–817. https://doi.org/10.1016/j.energy.2018.12.172
44. A. Aminmahalati, A. Fazlali, H. Safifikhani, Multi-objective optimization of CO boiler combustion chamber in the RFCC unit using NSGA II algorithm, *Energy*, **221** (2021), 119859. https://doi.org/10.1016/j.energy.2021.119859
45. P. Tan, J. Xia, C. Zhang, Q. Fang, G. Chen, Modeling and reduction of NO_X emissions for a 700MW coal-fired boiler with the advanced machine learning method, *Energy*, **94** (2016), 672–679. https://doi.org/10.1016/j.energy.2015.11.020
46. Q. Li, Q. He, Z. Liu, Low NO_x combustion optimization based on partial dimension opposition-based learning particle swarm optimization, *Fuel*, **310** (2022), 122352. https://doi.org/10.1016/j.fuel.2021.122352
47. H. Xi, P. Liao, X. Wu, Simultaneous parametric optimization for design and operation of solvent-based post-combustion carbon capture using particle swarm optimization, *Appl. Therm. Eng.*, **184** (2021), 116287. https://doi.org/10.1016/j.applthermaleng.2020.116287
48. M. V. J. J. Suresh, K. S. Reddy, A. K. Kolar, ANN-GA based optimization of a high ash coal-fired supercritical power plant, *Appl. Energy*, **88** (2011), 4867–4873. https://doi.org/10.1016/j.apenergy.2011.06.029
49. A. A. M. Rahat, C. Wang, R. M. Everson, J. E. Fieldsend, Data-driven multi-objective optimization of coal-fired boiler combustion systems, *Appl. Energy*, **229** (2018): 446–458. https://doi.org/10.1016/j.apenergy.2018.07.101
50. F. Wang, S. Ma, H. Wang, Y. Li, Z. Qin, J. Zhang, A hybrid model integrating improved flower pollination algorithm-based feature selection and improved random forest for NO_X emission estimation of coal-fired power plants, *Measurement*, **125** (2018), 303–312. https://doi.org/10.1016/j.measurement.2018.04.069

51. X. Hu, P. Niu, J. Wang, X. Zhang, Multi-objective prediction of coal-fired boiler with a deep hybrid neural networks, *Atmos. Pollut. Res.*, **11** (2020), 1084–1090. https://doi.org/10.1016/j.apr.2020.04.001
52. G. Huang, Q. Zhu, C. Siew, Extreme learning machine: a new learning scheme of feedforward neural network, in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, **2** (2004), 985–990. https://doi.org/10.1109/IJCNN.2004.1380068
53. Y. Ma, C. Xu, H. Wang, R. Wang, S. Liu, X. Gu, Model NO_x, SO_2 emissions concentration and thermal efficiency of CFBB based on a hyper-parameter self-optimized broad learning system, *Energies*, **15** (2022), 7700. https://doi.org/10.3390/en15207700

AIMS Press