A novel chaotic teaching learning based optimization algorithm and its application in optimization of extreme learning machine

Yunpeng Ma*, Haoheng Tang, Heqi Wang, Zhenying Wang, Xinxin Zhang, Lipeng Li

School of Information Engineering, Tianjin University of Commerce, Beichen, Tianjin 300134, China

Abstract: Recent ten years, the teaching learning based optimization algorithm (TLBO) has been widely concerned and successfully applied to solve various constraints and non-constraints problems. However, its convergence accuracy and convergence speed should be further improved. Therefore, a novel chaotic teaching learning based optimization algorithm (called CTLBO) is proposed. Firstly, chaotic variables are applied to initialize population individuals for increasing the diversity of population. Secondly, a kind of self-adaptive acceleration coefficient is introduced into teaching phase to enhance the convergence speed and solution quality. Finally, two population updating mechanisms are proposed to balance the exploration and exploitation capabilities in the learning phase. One is neighbor elitist search mechanism, another is chaos optimization mechanism. The performance of CTLBO is compared with five state-of-the-art optimization algorithms by several CEC mathematical problems. The experiment results show that the CTLBO yields better convergence rate than other algorithms on most testing functions. Additionally, the proposed CTLBO is applied to optimize the model parameters of extreme learning machine (ELM) and the tuned ELM is adopted to establish the NOx emissions model. Experiment results reveal that the NOx emissions model has good accuracy and meets the engineering requirement.

Keywords: Optimization; Model; Teaching Learning Based Optimization Algorithm; Chaos Theory; Extreme Learning Machine

1 Introduction

Lots of real-world optimization problems have complex properties, including high dimensional, multi-peak and non-conductive, so they are difficult to be solved by using traditional optimization techniques, such as steepest decent, dynamic programming and linear programming. For instance, if one optimization method requires gradient information of the problem, it can not solve non-conductive problem. Therefore, these exact optimization methods fail to provide an optima solution for complex optimization problems[1]. In order to address complex problems, many experts have proposed all kinds of efficient nature-inspired meta-heuristic optimization techniques, such as genetic algorithm[2], particle swarm optimization[3], artificial bee colony[4], ant colony
optimization algorithm[5], Krill herds algorithm[6], Social-spider optimization algorithm[7], Grey wolf optimization algorithm[8] and Teaching-learning-based optimization[9] etc.

Teaching learning based optimization algorithm (TLBO) was proposed to obtain global solutions for continuous non-linear functions and engineering optimization problems. It possesses some superior properties, such as high consistency, simple algorithm principle and less setting parameters. The TLBO algorithm has been applied to solve many of real-world optimization problems, such as electrical engineering [10-13], manufacturing processes[14-15] and economic load dispatch[16]. However, many improved TLBO algorithms were proposed to enhance the convergence speed and solution quality. In order to improve the solution quality of TLBO, Li et al [17] proposed a sort of ATLBO algorithm, which can solve several mathematics problems effectively. In [18], the elitism mechanism was introduced into TLBO to enhance its convergence performance. Several improved mechanisms were proposed to enhance the exploration and exploitation capacities of TLBO[19]. Mutation and crossover operators of differential evolution algorithm were introduced into TLBO, which can improve the exploration ability and increase the population diversity[20]. Huang et al. combined TLBO with cuckoo algorithm to enhance the local search ability of TLBO algorithm[21].Tuo et al. combined harmony search algorithm with TLBO algorithm for solving complex high-dimensional optimization problems[22]. These TLBO variants have shown faster convergence speed and better convergence accuracy than the original TLBO. And these TLBO variants have been successfully solved many real-world optimization problems.

Although the TLBO shows good performance, it still needs to be further improved. According to study the mechanism of TLBO, it has excellent exploration ability and poor exploitation ability. In addition, the convergence accuracy and solution quality are also enhanced simultaneously. Therefore, this paper proposes a novel chaotic teaching learning based optimization algorithm, which is called CTLBO. Firstly, the chaotic variables are applied to initialize population individuals for increasing the diversity of individuals. Secondly, a kind of self-adaptive acceleration coefficient is introduced into the teaching phase to increase the convergence speed and solution quality. In learning phase, two population individuals updating mechanisms are proposed to balance the exploration and exploitation capabilities. One is neighbor elitist search mechanism, another is chaos optimization mechanism. The detailed description of CTLBO is presented in the section 3.

To verify the performance of CTLBO, it is used to solve several CEC benchmark testing mathematics problems. The performance of CTLBO is compared to some state-of-the-art heuristic optimization algorithms. Experiment results show that the CTLBO yields higher convergence accuracy and better solution quality than other algorithms on most testing mathematics problems. Moreover, the CTLBO is applied to optimize the structure parameters of extreme learning machine (ELM)[23], and the tuned ELM is adopted to build the NOx emissions concentration model of one circulation fluidized bed boiler. Furthermore, the NOx emissions model has good accuracy and meets the engineering requirement.

The main contributions can be summarized as follows:

First, according to study the mechanism of TLBO, this paper proposes a kind of chaotic teaching learning based optimization algorithm. The chaotic variables are used to initialize the population individuals. Moreover, the neighbor elitist search mechanism and chaos optimization mechanism are introduced into the learning phase.

Second, several benchmark testing mathematics problems are applied to verify the performance of CTLBO. Compared to other excellent algorithms, CTLBO can find competitive
solutions on most testing problems and show higher convergence accuracy.

Third, the proposed CTLBO is used to optimize the input weights and bias of extreme learning machine, which can improve the model precision of ELM. And then, the tuned ELM is used to establish the NOx emissions concentration model.

The paper is organized as follows: we briefly review the basic knowledge of TLBO in section 2; we describe the proposed CTLBO algorithm in details in section 3; Section 4 shows the performance evaluation of the CTLBO; optimize the NOx emissions model by CTLBO in section 5; the conclusion is outlined in section 6.

2 Teaching learning based optimization algorithm

The TLBO algorithm is a kind of meta-heuristic population based intelligent algorithm, which was inspired from the mechanism of teaching and learning. TLBO algorithm has some advantages, such as easily understanding, high convergence precision and less tuning parameters. The process of TLBO algorithm consists of two phases: “Teaching phase” and “Learning phase”.

2.1 Teaching Phase

In teaching phase, learners gain knowledge from a teacher. The teacher is regarded as the most knowledgeable person in a class, and he can bring learners up to his level. Based on the mechanism, the teacher will try his best to make sure that the mean value of a class can keep up with his level. Supposed, at any iteration $i$, the algorithm calculates the mean value $M_i$ of marks of a class, and $M_{\text{new}}$ is the teacher. The teacher puts effort to enhance the mean $M_i$ up to his own level. All learners update their knowledge based on the following equation:

$$X_{\text{new},i} = X_{\text{old},i} + r_i (M_{\text{new}} - T_p M_i)$$

where $X_{\text{old},i}$ is the $i$th learner’s mark before learning from teacher, $X_{\text{new},i}$ is the mark after learning from teacher. $T_p$ is a teaching factor, $r_i$ varies from 0 to 1.

2.2 Learning Phase

In learning phase, learners enhance their knowledge through learning interaction among themselves. A learner interacts randomly with other learners to improve his or her knowledge via several ways, including group discussions, presentations, formal communications. A learner learns something new from the more knowledgeable person than him or her. The learning phase process can be described as follows:

At any iteration $i$, we randomly select two learners $X_i$ and $X_j$, where $i \neq j$.

$$X_{\text{new},i} = \begin{cases} 
  X_i + r_i (X_j - X_i) & \text{if } f(X_i) < f(X_j) \\
  X_j + r_i (X_j - X_i) & \text{if } f(X_j) < f(X_i) 
\end{cases}$$

(2)

The $X_{\text{new},i}$ is accepted if it owns better fitness function value.

3 Chaotic teaching learning based optimization algorithm

In this section, a novel chaotic teaching learning based optimization algorithm (namely CTLBO) is proposed, which can show better performance than the original TLBO algorithm. Compared with TLBO algorithm, the CTLBO algorithm has three excellent improvements. The chaotic variables are adopted to initialize population individuals, increasing the diversity of individuals. Based on literature [17], a self-adjusting acceleration coefficient is introduced into teaching phase to enhance convergence speed and solution quality. The most important improvement is that this paper proposes
two population updating mechanisms in the learning phase: one is neighbor elitist search mechanism another is chaos optimization mechanism, which can balance the exploration and exploitation capabilities. The CTLBO algorithm is described detailed as follows.

3.1 Population initialization
For TLBO and most TLBO variants, the initial population individuals are generated randomly, which maybe cause individuals in-homogeneity and affect the algorithm performance. Chaos theory has become a research hotpot in recent decade. Chaotic variables possess some properties, such as ergodicity, randomness and regularity. Based on merits of chaotic variables, they are adopted to initialize population individuals. The way of population initialization can keep the diversity of population and promote algorithm performance effectively. The most popular logistic mapping function is applied to initialize population individuals. The logistic mapping function is expressed as follows:

\[ y(t+1) = 4 \times y(t) \times (1 - y(t)) \] (3)
\[ x_d = \frac{1}{2} [u_d (1 + y_d) + l_d (1 - y_d)] \] (4)

Where \(-1 \leq y(t) \leq 1\), \( t = 1, 2, \cdots, ps \), ps is the population size. \( u_d \) is the upper limit value. \( l_d \) is the lower limit value. \( d \) is the solution dimension.

Firstly, an arbitrary solution is generated randomly. And based on equation (3), the remaining solutions are produced orderly. Finally, those solutions are mapped from chaotic space to search space based on equation (4).

3.2 Teaching phase
In this phase, a learner updates his or her marks through the previous mark \( X_{old,i} \) and the difference \((M_{new,Tj} - M_i)\). If \( X_{old,i} \) is very low, there is a big gap between the teacher and the learner. So a big correction is needed to improve the learner’s mark. Oppositely, if the obtained mark is high, only a small alteration is needed. In order to enhance the solution quality, we introduce a self-adaptive acceleration coefficient[17] into the mark update equation. Seen from equation (5), with the running of algorithm, the teacher plays an important role seriously on every iteration.

\[ X_{new,i} = \phi X_{old,i} + (1 - \phi) (M_{new,Tj} - M_i) \] (5)
\[ \phi = 1/(1 - \exp(-f(i)/fit) \times ni) \] (6)

where \( \phi \) is the acceleration coefficient, \( \phi \in [0,1] \), which decides the maximum step size. \( fit \) is the maximum fitness value in the first iteration. \( ni \) is the current iteration. \( f(i) \) is the current fitness value for \( ith \) learner.

3.3 Learning phase
In this phase, a kind of neighbor elitist search mechanism is introduced into the TLBO to avoid trapping into local optimum. In addition, the chaos optimization mechanism is introduced to balance the exploration and exploitation ability. The description of the two mechanisms are presented detailed.

3.3.1 neighbor elitist search mechanism. Based on the real teaching-learning scene, the neighbor elitist search mechanism is proposed. Before introducing the mechanism, a number of statements are needed. After teacher phase, the best student is selected as an elitist solution, noted \( X_{best} \). And we calculate the Euclidean distances between all the learners and the elitist solution. For instance,
the Euclidean distance between \( X_{\text{best}} \) and \( X_i \) is denoted as \( d_{\text{dis}_i} \). The Euclidean distance between \( X_{\text{best}} \) and \( X_{\text{mean}} \) is denoted as \( d \). Supposed that, if \( d_{\text{dis}_i} > d/5 \), the \( i \)th learner is regarded as the poor student, so he mainly obtains the knowledge from the teacher or the better student than him. For good student, he can explore his ability near the teacher. Noted: through trial and error, the judge condition is determined by if \( d_{\text{dis}_i} > d/5 \).

Supposed that there is a big gap between elitist individual and the \( i \)th student, a large-step search rule is needed to accelerate the scouting speed. Conversely, if there is a small gap between elitist individual and the \( i \)th student. A small-step seek is needed to improve the scouting accuracy.

According to the above description, the pseudo code of neighbor elitist search mechanism is presented as follows.

\[
\text{Pseudo code of learner phase} \\
\text{if } d_{\text{dis}_i} > d / 5 \\
\quad \text{if } \text{rand} < p \\
\quad \quad X_{\text{new}, i} = X_{\text{old}, i} + \psi_i (X_{\text{best}} - X_{\text{old}, i}) \\
\quad \text{else} \\
\quad \quad X_{\text{new}, i} = X_{\text{old}, i} + \text{rand} \times (X_j - X_{\text{old}, i}) \\
\text{end} \\
\text{else} \\
\quad \quad X_{\text{new}, i} = X_{\text{old}, i} + \| X_{\text{best}} - X_{\text{old}, i} \|_2 \times \text{rand} \\
\text{end} \\
\psi_i = 1/(1 + \exp(-f(i))) \\
\delta_i = \text{rand} \times \frac{1 - 1/(1 + \exp(-f(i)))}{\exp(-\min(f))}
\]

A selected probability value is generally set \( p = 0.5 \). Seen from the pseudo code, a learner can update his marks mainly depend on the elitist individual. Therefore, the proposed neighbor elitist search mechanism can improve the solution quality.

### 3.3.2 Chaos optimization mechanism

The conventional TLBO exists an obvious drawback. It is easy to reduce the population diversity and exploitation ability at latter stage. Therefore, the chaos optimization mechanism is introduced into the learner phase to balance the exploration and exploitation ability and increase the population diversity. As for the proposed mechanism, after teaching phase, a series of chaotic variables are generated based on the best individual as the new population individuals. And the mechanism can increase diversity of individuals. Based on the above explanation, the flowchart of CTLBO is shown in Figure 1.
Initialize population individuals based on chaos variables, set termination criterion

Calculate the mean of each design variables, calculate fitness values, choose best solution as a teacher

Modify population solutions based on Eq.(5)

Is new solution better than exiting?

Save the updated solutions via teacher phase

Yes

No

Adopt chaos optimization algorithm to update population individuals

Recalculate the mean of each design variables and fitness values, choose best solutions

Calculate the distance between best and mean value as d

Calculate the distance between best value and other population solution as dis

Update solution based on pseudo code of learner phase

Is new solution better than exiting?

Reject

Accept

Is termination criteria satisfied?

No

Yes

Find the best solution

Figure 1. the flow chart of CTLBO algorithm

Shown in Fig.1, after teacher phase, a threshold np is set to select the update mode. If the judge condition, \( \text{if}(\text{mod(\text{iteration,}np)==0}) \), is established, the chaos optimization mechanism is selected. Conversely, the neighbor elitist search mechanism is chosen. While the chaos optimization mechanism is selected, we only choose the best individual as the first chaotic variable after teacher phase. And then, the remaining individuals are generated based on equation (3). Finally, the chaotic individuals are converted from chaos space mapping to search space. In short, the new generated solutions are distributed in the range of global optimized solution, which can increase the diversity of populations and enhance the exploitation ability.

4 Experiment study

4.1 Performance testing

In this section, four uni-modal high dimension functions and four multi-modal high dimension
functions are selected to verify the performance of CTLBO algorithm. For uni-modal testing function, it has only one global optima, which can verify the convergence speed and exploitation ability of the CTLBO algorithm. For multi-modal testing function, it has multiple local solutions other than the global optima solution. So it is suitable to measure the exploration ability. These testing functions are given in table 1. Shown in table 1, the dimension of optimal parameters is denoted as \( n \), the limitation of optimal parameters is denoted as \( S \) and the desired global optima and optimum location are also presented. In addition, other heuristic optimization algorithms are chosen as the contrast methods, including ABC, GSA, TLBO, KH, SSO, which are the state-of-the-art optimization algorithms in recent years. All the experiments are carried out under Windows XP and Matlab 2014 with an Intel (R) Core(TM)64×2 Dual Core Processor T5670 @1.80GHz, 1.79GHz, 2GB RAM.

| Test functions | Global optima | Optimum location | S          |
|----------------|---------------|------------------|------------|
| \( F_1 \): Sphere function | 0 | \([0]^n\) | \([-100,100]^n\) |
| \( F_2 \): Schwefel 2.22 function | 0 | \([0]^n\) | \([-10,10]^n\) |
| \( F_3 \): Schwefel 1.2 function | 0 | \([0]^n\) | \([-100,100]^n\) |
| \( F_4 \): Schwefel function | 0 | \([0]^n\) | \([-100,100]^n\) |
| \( F_5 \): Ackley function | 0 | \([0]^n\) | \([-32,32]^n\) |
| \( F_6 \): Griewank function | 0 | \([0]^n\) | \([-600,600]^n\) |
| \( F_7 \): Penalized function | 0 | \([0]^n\) | \([-50,50]^n\) |
| \( F_8 \): Penalized 2 function | 0 | \([1]^n\) | \([-50,50]^n\) |

For the fairness of the test results, the population size is randomly set 40 and the maximum iteration number is 1000. The dimensions of all the functions are set 10, 50, 100, respectively. Parameters unique to the algorithms are set based on the original algorithm. The algorithm parameters are shown in table 2.

In this paper, two performance indexes (M: the mean of best solutions and S.D.: the standard deviation) are selected to verify the performance of six algorithms. The mean and S.D. are weaker, the performance of algorithm is better. In order to prove the repeatability of these algorithms, every experiment is repeated 30 times independently. The experiment results are represented from table 3 to table 5. The optimal solution is boldface.

| Method | Population | Iteration | Dimension | Other parameters |
|--------|------------|-----------|-----------|------------------|
| ABC    | 40         | 1000      | 10,50,100 | Limit=200        |
| GSA    | 40         | 1000      | 10,50,100 | \( G_0=100, \alpha=20 \) |
| SSO    | 40         | 1000      | 10,50,100 | --               |
| KH     | 40         | 1000      | 10,50,100 | --               |
| TLBO   | 40         | 1000      | 10,50,100 | --               |
| CTLBO  | 40         | 1000      | 10,50,100 | --               |
### Table 3. Performance comparison of 6 algorithms with 10 dimensions

| Method | Solution | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ | $F_8$ |
|--------|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| GSA    | M        | $2.3402 \times 10^{-18}$ | $4.8344 \times 10^{-9}$ | $6.6099 \times 10^{-9}$ | $1.1988 \times 10^{-9}$ | $2.4928 \times 10^{-9}$ | $4.490 \times 10^{-2}$ | $4.6695 \times 10^{-32}$ | $1.9497 \times 10^{-32}$ |
|        | S.D.     | $8.9715 \times 10^{-16}$ | $9.4608 \times 10^{-16}$ | $3.7773 \times 10^{-16}$ | $2.4542 \times 10^{-16}$ | $5.2221 \times 10^{-16}$ | $6.950 \times 10^{-16}$ | $1.5030 \times 10^{-32}$ | $3.1276 \times 10^{-32}$ |
| ABC    | M        | $1.1755 \times 10^{-17}$ | $3.2229 \times 10^{-17}$ | $16.5886 \times 10^{-17}$ | $0.0105 \times 10^{-17}$ | $7.9936 \times 10^{-17}$ | $0.748 \times 10^{-17}$ | $9.7452 \times 10^{-32}$ | $1.3645 \times 10^{-32}$ |
|        | S.D.     | $4.4411 \times 10^{-17}$ | $5.9858 \times 10^{-17}$ | $24.5983 \times 10^{-17}$ | $0.0047 \times 10^{-17}$ | $9.3299 \times 10^{-17}$ | $0.655 \times 10^{-17}$ | $4.5722 \times 10^{-32}$ | $4.1365 \times 10^{-32}$ |
| SSO    | M        | $3.5 \times 10^{-17}$ | $0.1162 \times 10^{-17}$ | $0.0047 \times 10^{-17}$ | $0.0330 \times 10^{-17}$ | $0.0610 \times 10^{-17}$ | $5.440 \times 10^{-17}$ | $1.4964 \times 10^{-17}$ | $1.180 \times 10^{-18}$ |
|        | S.D.     | $1.3233 \times 10^{-18}$ | $8.4690 \times 10^{-18}$ | $8.8219 \times 10^{-18}$ | $2.8230 \times 10^{-18}$ | $0.000 \times 10^{-18}$ | $0.000 \times 10^{-18}$ | $2.7568 \times 10^{-18}$ | $8.641 \times 10^{-18}$ |
| KH     | M        | $0$ | $14.646 \times 10^{-16}$ | $20.048 \times 10^{-16}$ | $8.8818 \times 10^{-16}$ | $0.000 \times 10^{-16}$ | $0.000 \times 10^{-16}$ | $2.5390 \times 10^{-16}$ | $0.9692 \times 10^{-16}$ |
|        | S.D.     | $0$ | $9.7146 \times 10^{-16}$ | $6.8061 \times 10^{-16}$ | $0.000 \times 10^{-16}$ | $0.000 \times 10^{-16}$ | $0.000 \times 10^{-16}$ | $0.0230 \times 10^{-16}$ | $0.0073 \times 10^{-16}$ |
| TLBO   | M        | $2.459 \times 10^{-23}$ | $1.225 \times 10^{-17}$ | $3.739 \times 10^{-17}$ | $1.5043 \times 10^{-17}$ | $4.677 \times 10^{-17}$ | $1.480 \times 10^{-17}$ | $2.4083 \times 10^{-17}$ | $0.0102 \times 10^{-17}$ |
|        | S.D.     | $0$ | $4.337 \times 10^{-17}$ | $2.3683 \times 10^{-17}$ | $4.2764 \times 10^{-17}$ | $9.0135 \times 10^{-17}$ | $1.790 \times 10^{-17}$ | $2.0540 \times 10^{-17}$ | $0.0189 \times 10^{-17}$ |
| CTLBO  | M        | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $8.8818 \times 10^{-16}$ | $0.7526 \times 10^{-16}$ |
|        | S.D.     | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0.6849 \times 10^{-16}$ |

### Table 4. Performance comparison of 6 algorithms with 50 dimensions

| Method | Solution | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ | $F_8$ |
|--------|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| GSA    | M        | $1.5457 \times 10^{-16}$ | $0.1881 \times 10^{-6}$ | $1.3771 \times 10^{-6}$ | $6.6514 \times 10^{-6}$ | $7.2302 \times 10^{-6}$ | $2.18 \times 10^{-6}$ | $21.80 \times 10^{-6}$ | $0.8685 \times 10^{-6}$ | $0.1403 \times 10^{-6}$ |
|        | S.D.     | $5.3958 \times 10^{-17}$ | $0.4780 \times 10^{-17}$ | $3.0584 \times 10^{-17}$ | $1.5131 \times 10^{-17}$ | $1.5446 \times 10^{-17}$ | $6.050 \times 10^{-17}$ | $49.2 \times 10^{-17}$ | $0.5594 \times 10^{-17}$ | $0.5636 \times 10^{-17}$ |
| ABC    | M        | $5.3930 \times 10^{-6}$ | $0.0028 \times 10^{-4}$ | $4.1063 \times 10^{-4}$ | $54.7144 \times 10^{-4}$ | $0.0350 \times 10^{-4}$ | $0.001 \times 10^{-4}$ | $0.000 \times 10^{-4}$ | $0.000 \times 10^{-4}$ | $0.000 \times 10^{-4}$ |
|        | S.D.     | $1.1994 \times 10^{-5}$ | $0.0010 \times 10^{-3}$ | $6.9442 \times 10^{-3}$ | $4.2383 \times 10^{-3}$ | $0.0279 \times 10^{-3}$ | $0.000 \times 10^{-3}$ | $0.000 \times 10^{-3}$ | $0.000 \times 10^{-3}$ | $0.000 \times 10^{-3}$ |
| SSO    | M        | $0.5654 \times 10^{-10}$ | $4.0976 \times 10^{-10}$ | $132.458 \times 10^{-10}$ | $3.1154 \times 10^{-10}$ | $3.6012 \times 10^{-10}$ | $0.000 \times 10^{-10}$ | $0.000 \times 10^{-10}$ | $0.000 \times 10^{-10}$ | $0.000 \times 10^{-10}$ |
|        | S.D.     | $2.2584 \times 10^{-15}$ | $2.7101 \times 10^{-15}$ | $8.6723 \times 10^{-15}$ | $1.3550 \times 10^{-15}$ | $1.8067 \times 10^{-15}$ | $0.000 \times 10^{-15}$ | $0.000 \times 10^{-15}$ | $0.000 \times 10^{-15}$ | $0.000 \times 10^{-15}$ |
| KH     | M        | $0$ | $0$ | NaN | $40.894 \times 10^{-6}$ | $8.8818 \times 10^{-6}$ | $0.000 \times 10^{-6}$ | $1.4570 \times 10^{-6}$ | $4.9055 \times 10^{-6}$ |
|        | S.D.     | $0$ | $0$ | NaN | $6.9885 \times 10^{-6}$ | $0.000 \times 10^{-6}$ | $0.000 \times 10^{-6}$ | $0.000 \times 10^{-6}$ | $0.000 \times 10^{-6}$ |
| TLBO   | M        | $9.0199 \times 10^{-28}$ | $6.8929 \times 10^{-28}$ | $6.1890 \times 10^{-28}$ | $2.1841 \times 10^{-28}$ | $5.8620 \times 10^{-28}$ | $0.000 \times 10^{-28}$ | $3.8029 \times 10^{-28}$ | $0.1623 \times 10^{-28}$ |
Seen from Table 3, we can find that the CTLBO obtains the global optimal solutions on 6 functions (F1, F2, F3, F4, F5, F6). For function F7, the performance of CTLBO is better than the conventional TLBO. However, for F8, the CTLBO cannot find the best function solution. Compared to other algorithms, the CTLBO owns better performance and higher convergence precision on most testing functions. Seen from Table 4 and Table 5, the proposed CTLBO still keeps the better performance than other optimization algorithms on most testing functions (F1, F2, F3, F4, F5, F6).

Table 5. Performance comparison of 6 algorithms with 100 dimensions

| Method | Solution | F1   | F2   | F3   | F4   | F5   | F6   | F7   | F8   |
|--------|----------|------|------|------|------|------|------|------|------|
| GSA    | M        | 192.75 × 10^3 | 3.0308 | 6.5847 × 10^3 | 12.646 × 10^3 | 2.0534 × 10^3 | 74.823 × 10^3 | 3.0752 × 10^5 | 5.3708 × 10^5 |
| S.D.   |          | 115.678 × 10^2 | 1.9809 | 1.6387 × 10^2 | 1.5837 × 10^2 | 0.4486 × 10^2 | 10.654 × 10^2 | 0.7362 × 10^2 | 24.980 × 10^2 |
| ABC    | M        | 0.0026 | 0.0786 | 1.7901 × 10^4 | 82.370 × 10^4 | 3.9347 × 10^4 | 0.1136 | 2.1459 × 10^4 | 1.6704 × 10^4 |
| S.D.   |          | 0.0016 | 0.0129 | 1.8125 × 10^3 | 2.8693 × 10^3 | 0.5896 × 10^3 | 0.1005 | 2.9909 × 10^3 | 5.3839 × 10^3 |
| SSO    | M        | 9.4303 | 27.302 | 2.4115 × 10^3 | 6.8200 | 3.6012 | 0.2963 | 3.4214 | 0.064 |
| S.D.   |          | 3.6134 | 10^15 | 7.2269 | 9.2504 | 4.5168 | 1.8067 | 1.6938 | 2.2584 | 4.4109 |
| KH     | M        | 5.206× 10^120 | 8.677× 10^120 | 2.7768 | 1.2723 | 5.9804 | 0.0028 | 3.8933 | 0.2290 |
| S.D.   |          | 5.3206 | 10^119 | 4.334 | 5.0289 | 1.7906 | 0.0152 | 2.1871 | 1.0482 |
| CTLBO  | M        | 0.0026 | 0.0786 | 1.7901 × 10^4 | 82.370 × 10^4 | 3.9347 × 10^4 | 0.1136 | 2.1459 × 10^4 | 1.6704 × 10^4 |
| S.D.   |          | 0.0016 | 0.0129 | 1.8125 × 10^3 | 2.8693 × 10^3 | 0.5896 × 10^3 | 0.1005 | 2.9909 × 10^3 | 5.3839 × 10^3 |


Compared with TLBO, the CTLBO obtains better performance on solutions quality and algorithmic stability on most testing functions, that is to say, the repeat-ability and robustness of the CTLBO are better than the original TLBO. Especially, according to the experiment results, the convergence speed and running time are faster than the conventional TLBO obviously. Therefore, the CTLBO has enhanced the global search ability and increased the convergence speed.

In order to see the comparative results clearly, we present several simulation diagrams, shown in Fig.2-Fig.7. The green solid line is the result of the CTLBO algorithm. Seen from Fig.2 to Fig.7, the CTLBO algorithm shows better average convergence rate than other comparative algorithms. Therefore, it is concluded that CTLBO is more efficient and more robustness.

Figure 2. Convergence rate comparison of F1 with dim=50
Figure 3. Convergence rate comparison of F2 with dim=50
Figure 4. Convergence rate comparison of F3 with dim=10
Figure 5. Convergence rate comparison of F4 with dim=50
Figure 6. Convergence rate comparison of F5 with dim=50
Figure 7. Convergence rate comparison of F6 with dim=50
5 Real world applications
In this section, the proposed CTLBO is used to optimize the NOx emission model of one 330MW circulation fluidized bed boiler. Recently, industrial and civilian electricity are mainly derived from the thermal power generation across the world. Therefore, the power plants have consumed large amount of coal resources and emitted lots of polluting gases into the air. Energy saving and emission reduction is still the theme of the times. We must be paid highly attention. So it is profound necessary to optimize the boiler combustion operation process to improve the thermal efficiency and reduce the polluting gas emission. In recent years, a number of scholars and experts have proposed many methods to settle the boiler combustion optimization problem[24-29].

As for the boiler combustion optimization problem, it is essential to reduce NOx emissions. Before reducing NOx emission, a relative accurate NOx emission model must be established. However, the combustion process of boiler possess several complex properties, such as large lag, sluggishness and non-linearity. Therefore, it is profound difficult to build NOx emissions model based on the combustion mechanism. To solve the problem, we adopt the tuned extreme learning machine by the CTLBO algorithm to build NOx emissions model. Actually, the CTLBO is applied to optimize the input weights and bias of ELM, which is called CTLBO-ELM. The detailed process is shown as follows.

5.1 Tuned extreme learning machine
For extreme learning machine, the input-weights and bias of hidden layers are randomly generated. And the output layer weights matrix are analytically calculated based on the least square method, which is unique. However, the weights of input layer and thresholds of hidden layer may be not optimal parameters. Therefore, in order to find the optimal weights of input layer and thresholds of hidden layer, this paper uses the proposed CTLBO to optimize the model parameters of extreme learning machine.

Supposed that the ELM has $n$ input nodes and $m$ hidden nodes, so the dimension of the optimized parameters is $(n+1) \times m$ and the $i$th optimized individual is denoted as $\alpha_i=[\omega_{i1}, \omega_{i2}, \cdots, \omega_{in}, b_1, \cdots, b_m]$ , where $\omega_{im} \in [-1,1]$ and $b_i \in [0,1]$ .Additionally, The fitness function is set as follows.

$$f(\alpha_i) = \sqrt{\frac{\sum_{j=1}^{N_{train}} \left( \sum_{i=1}^{m} \beta_i g(\omega_i x_j + b_i) - t_j \right)^2}{m \times N_{train}}} \tag{9}$$

Shown in Eq.(9), $g(\cdot)$ is the hidden layer function, $t_j$ is the $j$th target value, $N_{train}$ is the number of training data.

The calculation process of tuned ELM is presented as follows.
1) Assign the dimension of the optimized individual based on the number of input nodes and hidden nodes.
2) Parameters initialization of the CTLBO, such as population size, maximum iteration number and terminal condition.
3) Assign fitness function and calculate fitness function values. The best individual is regarded as teacher.
4) The iterative optimization: teaching phase and learning phase.
5) Stop the optimization and save the best input weights and bias.
6) Based on the best input weights and bias, calculate the output weights.

5.2 Model NOx emissions
In order to verify the validity of the tuned ELM, it is used to establish the NOx emissions model. In this paper, 300 group combust ion operation cases data of one 330MW CFBB are used to build the NOx emissions model. And these operation data are collected from literature [29]. Therefore, the
detailed description of these data can be found in the literature [29].

This paper uses the tuned ELM to establish NOx emissions model, which represents the function mapping relation between NOx emissions and 20 combustion operational conditions. Actually, the 20 operational parameters are regarded as the input value and the NOx emission concentration is set as the output value. Here, the objective function of CTLBO is set as $\min \sqrt{\frac{1}{n} \sum (g(x_i) - t_i)^2}$, where $g(x_i)$ is the prediction output value, $t_i$ is the target output value, $n$ is the number of training or testing sample. The number of hidden neurons is set 20 and the hidden active function is set as ‘sigmoidal function’. In this paper, the objective function value is called as training accuracy or testing accuracy. For training model, if the training accuracy is small, the model owns favorable regression ability. For testing model, if the testing accuracy is small, the model possesses good generalization ability. If the training accuracy and testing accuracy are suitable for industrial requirement, the model is effective theoretically. If the model has small training accuracy and big testing accuracy, the model is over-fitting and inadvisable seriously.

For training data, the training accuracy of the CTLBO-ELM is 0.0812 but the ELM is 0.1073. The results show that the CTLBO-ELM owns better performance ability and high regression precision. For testing data, the testing accuracy of CTLBO-ELM is 0.0872, and the ELM is 0.1393. So the NOx emission model is effective and feasible. Shown in Fig.8, the curve of the CTLBO-ELM is better obviously than ELM. The output of CTLBO-ELM could be close to the target output except several testing points. Seen from Fig.9, the error curve of CTLBO-ELM is controlled in the range of [-10,10] except some points. For engineering, it is feasible. However, the error curve of ELM fluctuates remarkably. In conclusion, the CTLBO-ELM possesses more favorable identification ability and generalization ability than conventional ELM.

![Figure 8. comparison of predicting model output of NOx emissions](image1)

![Figure 9. Comparison of testing error between CTLBO-ELM with ELM](image2)

6 Conclusion
To improve the performance of conventional TLBO, this paper proposes a kind of chaotic TLBO. The chaotic variables are used to initialize population individuals. A kind of self-adaptive acceleration coefficient is introduced in the teaching phase. In learner phase, the neighbor elitist search mechanism and chaos optimization mechanism are introduced to balance the exploration and exploitation capabilities. Compared with several excellent optimization algorithms, the CTLBO algorithm presents better solution quality and higher convergence rate on several benchmark functions. In addition, the CTLBO algorithm is applied to optimize the input weights and bias of ELM, finding the best model parameters. And then, the optimized ELM is used to build NOx emissions concentration model. Experiment results reveal that the optimized ELM by CTLBO has good regression accuracy and
generalization ability.

In the future, we will focus on the following tasks:

Based on the CTLBO algorithm, a kind of multi-objective CTLBO algorithm will be designed to solve multi-objective optimization problems. The CTLBO algorithm will be further improved to address dynamic constrained optimization problems.

ACKNOWLEDGMENT

Funding: This work is supported by the National Natural Science Foundation of China (Grant No.61573306) and the National Science Foundation of Tianjin (Grant No.20JCQNJC00430) and the Technical Innovation Guide Foundation of Tianjin (Grant No. 20YDTPJC00320) and College Students’ Innovative Entrepreneurial Training Plan Program (Grant No.202010069066)

References

[1] Doğan B, Ölzeme T. A new metaheuristic for numerical function optimization: Vortex Search algorithm[J]. Information Sciences, 2015, 293:125-145.
[2] John, H, Holland. Genetic Algorithms and the Optimal Allocation of Trials[J]. Siam Journal on Computing, 1973, 2:88-105.
[3] Kennedy J. Particle swarm optimization[M]. Encyclopedia of Machine Learning. Springer US, 2010: 760-766.
[4] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. Journal of global optimization, 2007, 39(3): 459-471.
[5] Dorigo M, Birattari M, Thomas Stützle. Ant colony optimization[J]. IEEE Computational Intelligence Magazine, 2007, 1(4):28-39.
[6] Gandomi A H, Alavi A H. Krill herd: a new bio-inspired optimization algorithm[J]. Communications in Nonlinear Science and Numerical Simulation, 2012, 17(12): 4831-4845.
[7] Cuevas E, Cienfuegos M, Zaldivar D, et al. A swarm optimization algorithm inspired in the behavior of the social-spider[J]. Expert Systems with Applications, 2013, 40(16): 6374-6384.
[8] Mirjalili S, Saremi S, Mirjalili S M, et al. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization[J]. Expert Systems with Applications, 2016, 47:106-119.
[9] Rao R V, Savsani V J, Vakharia D P. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems[J]. Computer-Aided Design, 2011, 43(3): 303-315.
[10] Zhou Y, Zheng S. Climate adaptive optimal design of an aerogel glazing system with the integration of a heuristic teaching-learning-based algorithm in machine learning-based optimization[J]. Renewable Energy, 2020, 153:375-391.
[11] Niknam T, Azizipanah-Abarghoee R, Narimani M R. An efficient scenario-based stochastic programming framework for multi-objective optimal micro-grid operation[J]. Applied Energy, 2012, 99: 455-470.
[12] Rao R V, Pawar R B. Quasi-oppositional-based Rao algorithms for multi-objective design optimization of selected heat sinks[J]. Journal of Computational Design and Engineering, 2020, 7(6):830-863.
[13] Niknam T, Golestaneh F, Sadeghi M S. 0-Multiobjective Teaching–Learning-Based Optimization for Dynamic Economic Emission Dispatch[J]. Systems Journal, IEEE, 2012, 6(2): 341-352.
[14] Rao R V, Kalyankar V D. Parameter optimization of modern machining processes using teaching–learning-based optimization algorithm[J]. Engineering Applications of Artificial Intelligence, 2013, 26(1): 524-531.
[16] Krishnanand K R, Panigrahi B K, Rout P K, et al. Application of multi-objective teaching-learning-based algorithm to an economic load dispatch problem with incommensurable objectives[M]//Swarm, Evolutionary, and Memetic Computing. Springer Berlin Heidelberg, 2011: 697-705.

[17] Li G, Niu P, Zhang W, et al. Model NOx emissions by least squares support vector machine with tuning based on ameliorated teaching–learning-based optimization[J]. Chemometrics and Intelligent Laboratory Systems, 2013, 126: 11-20.

[18] Rao R, Patel V. Comparative performance of an elitist teaching-learning-based optimization algorithm for solving unconstrained optimization problems[J]. International Journal of Industrial Engineering Computations, 2013, 4(1): 29-50.

[19] Rao R V, Patel V. An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems[J]. Scientia Iranica, 2013, 20(3): 710-720.

[20] Yu K, Wang X, Wang Z. An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems[J]. Journal of Intelligent Manufacturing, 2016, 27(4):831-843.

[21] Huang J, Gao L, Li X. An effective teaching-learning-based cuckoo search algorithm for parameter optimization problems in structure designing and machining processes[J]. Applied Soft Computing, 2015, 36(C):349-356.

[22] Tuo S, Yong L, Deng F, et al. HSTLBO: A hybrid algorithm based on Harmony Search and Teaching-Learning-Based Optimization for complex high-dimensional optimization problems.[J]. Plos One, 2017, 12(4).

[23] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: a new learning scheme of feedforward neural networks[C]//Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. IEEE, 2004, 2: 985-990.

[24] Song J, Romero C E, Yao Z, et al. Improved artificial bee colony-based optimization of boiler combustion considering NOX emissions, heat rate and fly ash recycling for on-line applications[J]. Fuel, 2016, 172: 20-28.

[25] Rahat A A A M, Wang C L, Everson R M, et al. Data-driven Multi-objective Optimization of Coal-fired Boiler Combustion Systems[J]. Applied Energy, 2018, 229: 446-458.

[26] Niu P, Li J, Chang L, et al. A Novel Flower Pollination Algorithm for Modeling the Boiler Thermal Efficiency[J]. Neural Processing Letters, 2019, 49:737-759.

[27] Hu X, Niu P, Wang J, et al. Multi-objective prediction of coal-fired boiler with a deep hybrid neural networks - ScienceDirect[J]. Atmospheric Pollution Research, 2020, 11(7):1084-1090.

[28] Li X, Niu P, Liu J. Combustion Optimization of a Boiler Based on the Chaos and Lévy Flight Vortex Search Algorithm[J]. Applied Mathematical Modelling, 2018, 58:3-18.

[29] Ma Y, Zhang X, Song J, Chen L. A modified teaching–learning-based optimization algorithm for solving optimization problem[J]. Knowledge-Based Systems, 2021, 212:106599.