

Research Article

An Improved Teaching-Learning-Based Optimization Algorithm with Reinforcement Learning Strategy for Solving Optimization Problems

Di Wu ¹, Shuang Wang ², Qingxin Liu ³, Laith Abualigah ^{4,5,6} and Heming Jia ²

¹School of Education and Music, Sanming University, Sanming 365004, China

²School of Information Engineering, Sanming University, Sanming 365004, China

³School of Computer Science and Technology, Hainan University, Haikou 570228, China

⁴Research and Innovation Department, Skyline University College, Sharjah 1797, UAE

⁵Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan

⁶School of Computer Science, Universiti Sains Malaysia, George Town, Pulau Pinang 11800, Malaysia

Correspondence should be addressed to Shuang Wang; wang_shuang9279@163.com and Heming Jia; jiaheminglucky99@126.com

Received 21 August 2021; Revised 1 December 2021; Accepted 22 February 2022; Published 24 March 2022

Academic Editor: Navid Razmjooy

Copyright © 2022 Di Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an improved teaching-learning-based optimization (TLBO) algorithm for solving optimization problems, called RLTLBO. First, a new learning mode considering the effect of the teacher is presented. Second, the Q-Learning method in reinforcement learning (RL) is introduced to build a switching mechanism between two different learning modes in the learner phase. Finally, ROBL is adopted after both the teacher and learner phases to improve the local optima avoidance ability of RLTLBO. These two strategies effectively enhance the convergence speed and accuracy of the proposed algorithm. RLTLBO is analyzed on 23 standard benchmark functions and eight CEC2017 test functions to verify the optimization performance. The results reveal that proposed algorithm provides effective and efficient performance in solving benchmark test functions. Moreover, RLTLBO is also applied to solve eight industrial engineering design problems. Compared with the basic TLBO and seven state-of-the-art algorithms, the results illustrate that RLTLBO has superior performance and promising prospects for dealing with real-world optimization problems. The source codes of the RLTLBO are publicly available at <https://github.com/WangShuang92/RLTLBO>.

1. Introduction

In recent years, real-world optimization problems have become increasingly complex and diverse in a wide range of fields and disciplines. Traditional (mathematical) optimization methods, such as Newton's method and the gradient descent method can no longer meet the needs for solving current optimization problems. Thus, nontraditional methods, especially metaheuristic algorithms, are becoming increasingly pervasive among researchers [1–3]. Metaheuristics are algorithms based on intuition or experience, that can provide a feasible solution at an acceptable cost (referring to computing time and computational resources),

and the deviation between the feasible solution and the optimal solution may not be predicted in advance. Metaheuristic optimization algorithms have the merits of being flexible, having few parameters and avoiding local optima. Additionally, they can be rapidly deployed and thus have been utilized for solving various optimization problems over the past decades [4, 5]. Some of the most representative meta-heuristic algorithms are listed as follows: genetic algorithms (GA) [6], differential evolution algorithm (DE) [7], simulated annealing (SA) [8], arithmetic optimization algorithm (AOA) [9], heat transfer relation-based optimization algorithm (HTOA) [10], particle swarm optimization (PSO) [11], salp swarm algorithm (SSA) [12], grey wolf

optimizer (GWO) [13], whale optimization algorithm (WOA) [14], aquila optimizer (AO) [15], remora optimization algorithm (ROA) [16], etc.

Teaching-learning-based optimization (TLBO) is a meta-heuristic algorithm proposed by Rao et al. in 2011 [17]. The TLBO method is inspired by the teaching-learning process in a class and simulates the influence of a teacher on learners. Due to the advantages of rapid convergence, absence of algorithm-specific parameters and easy implementation, TLBO has become a viral optimization algorithm and has been successfully applied to real-world problems in diverse fields. Aouf et al. [18] applied TLBO to optimize the parameters of the ANFIS structure to obtain the optimal trajectory and traveling time to address the navigation problem of the mobile robot in a strange environment. Singh et al. [19] studied the application of TLBO for optimal coordination of directional overcurrent relays (DOCRs) in a looped power system. Multiobjective TLBO was applied to solve the motif discovery problem (MDP) in the bioinformatics field by Gonzalez-Alvarez et al. [20], and obtained better solutions than other biology-based multiobjective evolutionary algorithms. All the above applications have suggested that TLBO can be effectively applied to many optimization problems in various fields.

The improvement and hybrid algorithms of TLBO and their applications have also been studied by several researchers [21]. Kumar and Singh [22] developed a chaotic version of TLBO with different chaotic mechanisms. A local search method was also incorporated to guide the search direction between local and global search and to improve the quality of solution. The application of clustering problems proved the effectiveness of this algorithm. Taheri et al. [23] proposed a balanced TLBO with three modifications, called BTLBO. A weighted mean replaced the mean value in the teacher phase to maintain the diversity. The tutoring phase was added as a powerful local search mechanism for exploiting regions around the best solution. The restarting phase was introduced to improve the exploration ability by replacing inactive learners with randomly initialized learners. Ma et al. [24] proposed a modified TLBO (MTLBO) by introducing a population group mechanism into the basic TLBO. All students were divided into two groups and updated by different updating strategies. The MTLBO was also applied to establish the NO_x emission model of a circulation fluidized bed boiler. Xu et al. [25] introduced dynamic-opposite learning (DOL) strategy into TLBO to overcome premature convergence. The asymmetric search space and the dynamic change in the characteristics of DOL help DOLTLBO to holistically improve the exploitation and exploration capabilities. Dong et al. [26] presented a KTLBO algorithm to achieve computationally expensive constrained optimization. The kriging-assisted two-phase optimization framework was used to alternately conduct global and local searches, achieving the search acceleration. KTLBO was also adopted to design the structure of a blended-wing-body underwater glider. Ren et al. [27] developed a multiobjective elitist feedback TLBO (MEFTO) for multiobjective optimization problems. The elitism strategy was used to store the best solutions obtained thus far. The proposed feedback

phase allowed students to choose whether to study directly with the teacher or to motivate themselves, providing a novel way for students to improve themselves. Zhang et al. [28] proposed a hybrid algorithm based on TLBO and a neural network algorithm (NNA) named TLNNA to solve engineering optimization problems. The experimental results suggested that TLNNA has improved global search ability and fast convergence speed. By considering the features of the WOA and TLBO, Lakshmi and Mohanaiah [29] proposed a hybrid WOA-TLBO algorithm. This was also applied to solve the facial emotion recognition (FER) functional problem, and the reported results showed its effectiveness and high accuracy.

The TLBO variants proposed previously have improved searchability and accelerated the convergence process, but they still struggle with premature convergence and insufficient learning processes. Thus, in this paper we propose an improved TLBO algorithm to solve industrial engineering optimization problems. Given the characteristics of TLBO, reinforcement learning (RL) in machine learning is introduced to the learner phase, and enables the algorithm to choose a more suitable learning mode, which can train the search agents to perform more beneficial actions. In addition, a random opposition-based learning (ROBL) strategy is added after the whole learner phase to facilitate the convergence acceleration and avoid local optima. The proposed improved TLBO with RH and ROBL strategies is called RLTLBO. The standard and CEC2017 benchmark functions and eight engineering design problems are used to test the exploration and exploitation capabilities of the proposed method. The RLTLBO algorithm is compared with some existing algorithms, including the basic TLBO and the Salp Swarm Algorithm (SSA), which are considered the classical algorithms, the Aquila Optimizer (AO), Harris Hawks Optimization (HHO) [30], and Horse herd Optimization Algorithm (HOA) [31], which are the recent new methods, and the memory-based Grey Wolf Optimizer (mGWO) [32], modified Ant Lion Optimizer (MALO) [33] and dynamic Sine Cosine Algorithm (DSCA) [34], which are the latest improved algorithms. The experimental results show that the proposed RLTLBO method is superior to the state-of-the-art algorithms in exploration and exploitation capabilities. Moreover, eight industrial engineering design problems are applied to evaluate the effectiveness of the algorithm when solving real-world optimization problems.

The rest of this paper is organized as follows: Section 2 provides a brief overview of the basic TLBO, RL, and ROBL strategies. Section 3 describes the proposed RLTLBO algorithm in detail. Simulations, experiments and an analysis of the results are presented in Section 4. Section 5 describes industrial engineering design problems. Finally, Section 6 concludes the paper.

2. Related Work

2.1. Teaching-Learning-Based Optimization. The TLBO algorithm mimics the influence of a teacher on the output of learners, which can be reflected by learners' grades. As a highly learned person, the teacher gives their knowledge to

the learners. The outcome of the learners is affected by the quality of the teacher. It is obvious that learners trained by a good teacher can achieve better results in terms of their grades. The optimization process of TLBO is divided into two phases: the teacher phase and the learner phase.

2.1.1. Teacher Phase. The teacher phase simulates the teaching process of a teacher. The best one in the class is selected as the teacher, and then the teacher tries their best to improve the overall level of the class. The teaching process can be formulated as follows:

$$X_{\text{new}} = X_{\text{old}} + \text{rand}(X_{\text{teacher}} - T_F \cdot \text{Mean}), \quad (1)$$

where X_{new} and X_{old} represent the positions of the individual after and before learning, that is, the candidate solutions after and before updating. X_{teacher} is the position of the teacher, which is the best individual of the population. Mean indicates the average level of search agents in the population. T_F is a teaching factor that determines the change of the mean value, and rand is a random number between 0 and 1. The value of T_F can be either 1 or 2, which is a heuristic step and randomly decided with equal probability as $T_F = \text{round}(1 + \text{rand}(0, 1)\{2-1\})$.

2.1.2. Learner Phase. In addition to learning new knowledge from the teacher, learners can also increase knowledge through interaction. In the mutual learning process, a learner can randomly learn knowledge from another learner with a better grade randomly. The expression of the learner phase can be written as follows:

$$X_{\text{new}} = \begin{cases} X_{\text{old}} + \text{rand}(X_{r1} - X_{r2}) & f(X_{r1}) < f(X_{r2}) \\ X_{\text{old}} + \text{rand}(X_{r2} - X_{r1}) & \text{otherwise} \end{cases}, \quad (2)$$

where X_{r1} and X_{r2} indicate the positions of two learners randomly selected from the population. $f(\cdot)$ is the fitness value. The comparison between two learners determines the learning direction. The individual with a poor grade learns from the individual with a better grade. The new individual with improvements after learning will be accepted, otherwise rejected.

The flow chart of the TLBO algorithm is shown in Figure 1.

2.2. Reinforcement Learning (RL). Machine learning algorithms are also widely used to solve various optimization problems [35]. Machine learning methods generally consist of four categories, as shown in Figure 2: supervised learning, unsupervised learning, semisupervised learning, and reinforcement learning (RL). In RL algorithms, the agent is trained to learn optimal actions in a complex environment. The agent is trained in different ways and uses its training experience in the subsequent actions. RL methods generally consist of model-free and model-based approaches. The model-free approaches can be divided into two subgroups: value-based and policy-based methods. The value-based algorithms are convenient for coordinating with

meta-heuristic algorithms because they are model-free and policy-free, providing higher flexibility [36]. In the value-based RL approaches, the reinforcement agent learns from its actions and experience in the environment, such through reward and penalty. The agent measures the success of the action in completing the task goal through the reward penalty and then makes a decision based on its achievement.

The Q-Learning method is one of the representative algorithms among the value-based RL methods. In the Q-Learning method, the agent takes random actions and then obtains a reward or penalty. An experience is gradually constructed based on the agent's actions. Throughout process of building experience, a table called Q-Table is defined [37]. The agent considers all possible actions and tries to update its state according to the Q-Table values to select the best action that maximizes the current state's maximal rewards. Therefore, the agent in action determines whether to explore or exploit the environment.

Compared to RL methods, meta-heuristic algorithms often require deep expert knowledge to establish the balance between different phases. RL methods can help discover optimal designs of parameters and more balanced strategies allowing the algorithm to switch between the exploration and exploitation phases. Metaheuristic methods usually operate with specific policies in certain situations, and thus, the dynamism is lower than that of RL algorithms, especially value-based methods. The agent in the value-based methods is online and operates beneficial actions through a reward-penalty mechanism without following any policy. Many types of research have been presented in the literature regarding the combination of meta-heuristics and RL [38–44].

2.3. Random Opposition-Based Learning (ROBL). Random opposition-based learning (ROBL) is a variant of opposition-based learning (OBL) [45] proposed by Long et al. in 2019 [46]. OBL is a powerful optimization tool that simultaneously considers the fitness of an estimate and its corresponding opposite estimate to achieve a better candidate solution. In contrast from the basic OBL, ROBL utilizes a random term to improve the OBL strategy, which is defined as follows:

$$\hat{x}_j = l_j + u_j - \text{rand} \times x_j, \quad j = 1, 2, \dots, n, \quad (3)$$

where \hat{x}_j and x_j indicate the opposite and original solutions, u_j and l_j are the upper and lower bound of the problem in j th dimension. The opposite solution is randomly selected in the opposite half of the search space. This solution is not only opposite, but also random, with a wider range of distributions. An example of ROBL solutions is shown in Figure 3. The opposite solution with a random term described by equation (3) is more stochastic than the basic OBL and can effectively help the algorithm jump out of the local optima.

3. The Proposed RLTLBO Algorithm

3.1. New Learning Mode. The basic TLBO algorithm performs the learner phase after the teacher phase in each iteration. The search agent learns from other individuals in the

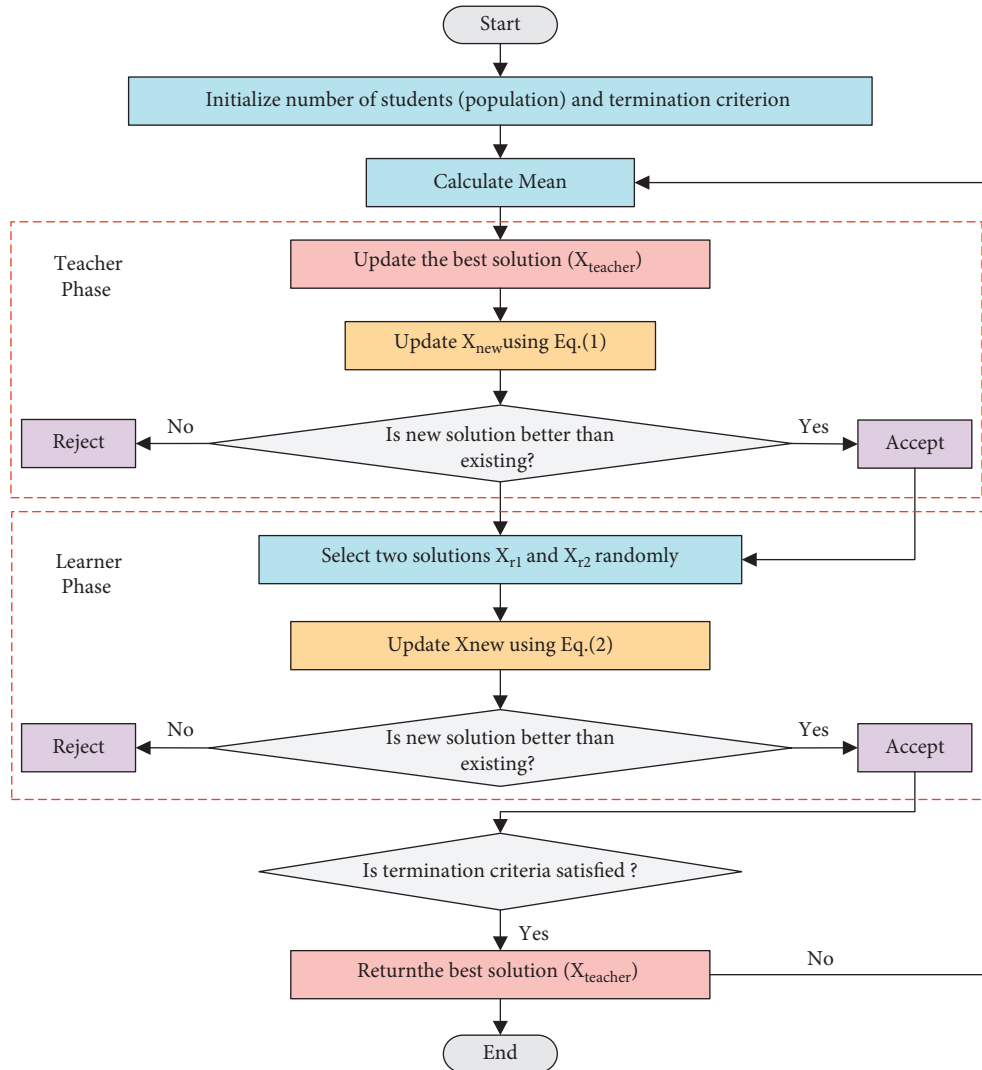


FIGURE 1: The flowchart of TLBO.

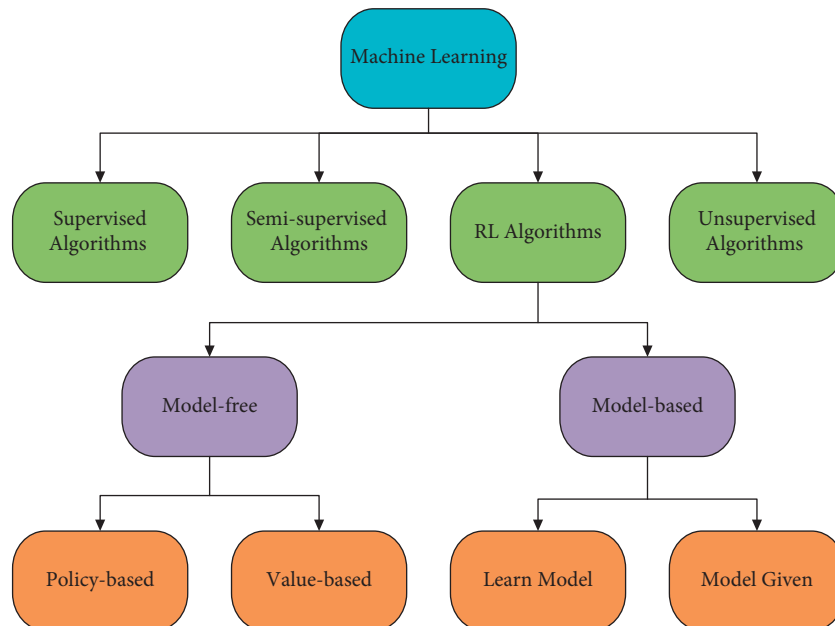


FIGURE 2: Classification of the reinforcement learning algorithms.

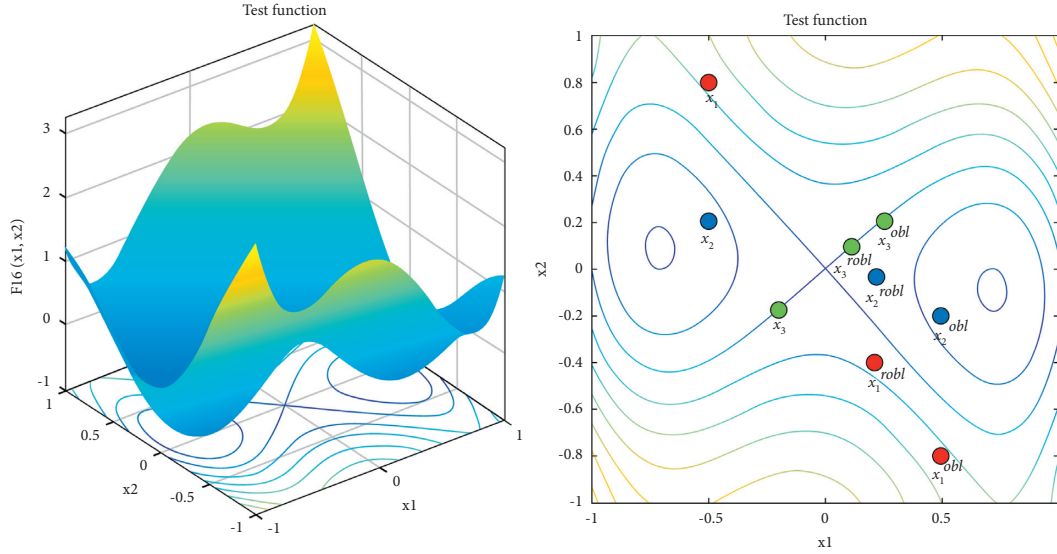


FIGURE 3: Example of ROBL solutions. Three sets of solutions (original solution, corresponding opposite solution (xobl), and random opposite solution (xrobl)) are labeled in a two-dimensional search space. The random opposite solutions are not only in the symmetric positions, but also with a wider range of distributions.

learner phase. However, in the actual learning process, students learning from each other varies from person to person. Different students might choose different learning modes, such as formal communications, group discussions, presentations, etc. Moreover, the students might adjust the

learning mode according to their learning situation during the learning process. Therefore, in this paper, we introduce another learning mode to diversify the learning methods of the students, which can be described in the equations as follows:

$$X_{\text{new}} = \begin{cases} X_{\text{old}} + \text{rand} \left[\left(1 - \frac{t}{T} \right) X_{r3} + \left(\frac{t}{T} \right) X_{\text{teacher}} - X_{\text{old}} \right] & f(X_{r3}) < f(X_{\text{old}}) \\ X_{\text{old}} + \text{rand} \left[\left(1 - \frac{t}{T} \right) X_{\text{old}} + \left(\frac{t}{T} \right) X_{\text{teacher}} - X_{r3} \right] & \text{otherwise} \end{cases}, \quad (4)$$

where X_{r3} is the position of a learner randomly selected from the population. t and T are the current and maximum number of iterations.

In this learning mode, the effect of the teacher is introduced. Sometimes the mutual learning between students is not always beneficial, and the partial intervention of the teacher is more helpful to students' improvement. Students will not only learn from each other but also ask the teacher for help. At the beginning of the iterations, the weight of mutual learning among students is larger, and the algorithm pays more attention to random learning, which can maintain population diversity and increase global searchability. In the later iteration stage, students consult more from the teacher and approach the teacher, enhancing the algorithms local searchability.

3.2. Learner Phase with RL Strategy. To enable students to adjust their learning mode more effectively, Q-Learning in RL is introduced to complete the switching between both learning modes. The student uses Q-Table values as a guide to decide between different learning modes. The Q-table is

updated using a reward-penalty mechanism. The student selects the best state by calculating the benefit degree of each possible state and taking the leaning mode with the highest Q-values for the next step. The student obtains a reward or a penalty according to its actions after each step. The general pattern of the RL agent and environmental framework is shown in Figure 4.

In the Q-Learning method, a reward table is used to reward or penalize the agent for its action or state compositions, which users can provide. The reward table in this work contains the positive (+1) or negative (-1) rewards for each state and action couple. The Q-Table can be considered the agents experience, which should be assigned a zero value for all units in the beginning. Consequently, the student updates Q-Table using the Bellman equation (5) and prepares the Q-Table for the next iteration [44].

$$Q_{(t+1)}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \lambda \left[r_{t+1} + \gamma \text{Max}_{(s_{t+1}, a)} Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \right], \quad (5)$$

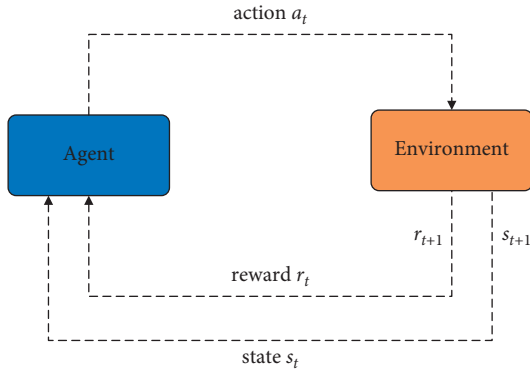


FIGURE 4: Reinforcement learning agent and environment framework. a_t represents the current action s_t and s_{t+1} indicate the current and the next state, r_t and r_{t+1} indicate the current and the next reward, respectively.

where s_t and s_{t+1} indicate the current and the next state respectively, Q_t and Q_{t+1} are the current Q-value and pre-estimated Q-value for the next state s_{t+1} , and a_t represents the current action. λ and γ are the learning rate value and discount factor, respectively, which are numbers between 0 and 1. The learning rate determines how fast the algorithm should learn and controls the convergence of the learning process. The discount factor defines how much the algorithm learns from the mistake and controls the importance of future rewards. r_{t+1} indicates the immediate reward or penalty an agent gets for taking current action.

In each iteration, the agent uses equation (5) to calculate and weight each possible state and action for the next step, before choosing the best action (learning mode 1 or learning mode (2) with the highest likelihood to get closer to the best optimal solution. Examples of the reward table and Q-Table are displayed in Figure 5. This RL strategy helps establish a switching mechanism between different learning modes in the learner phase and find the most suitable decision scheme. Four optional actions can occur as listed below:

- (1) When the student is learning in learning mode 1, they still decides to stay in learning mode 1
- (2) When the student is learning in learning mode 2, they still decides to stay in learning mode 2
- (3) When the student learns in learning mode 1, they decides to transition to learning mode 2
- (4) When the student learns in learning mode 1, they decides to transition to learning mode 2

The most important value of the RL strategy is to help the algorithm switch between different learning modes as and when needed during the learner phase. For the above reason, the algorithm can find better solutions faster and more effectively in the search space, considerably increasing the search efficiency. Therefore, the convergence speed of the algorithm can be improved effectively.

3.3. The Detail of RLTLBO. In the improved TLBO algorithm, the teacher phase of basic TLBO is carried out first. Then, the learner phase with RL strategy is implemented to

achieve effective and efficient investigation in the search space. Finally, ROBL is added to enhance the ability of local optima avoidability. The random opposite solution increases the probability of the algorithm finding a better solution. This variant of TLBO, which incorporates RL, is named RLTLBO. The pseudocode and the flowchart of the proposed RLTLBO algorithm are shown in Algorithm 1 and Figure 6, respectively.

3.4. Computational Complexity Analysis. RLTLBO mainly consists of three components: initialization, fitness evaluation, and position updating. In the initialization phase, the computational complexity of positions generated is $O(N)$. Then, the computational complexity of fitness evaluation for the solution is $O(2 \times N)$ during the iteration process. Finally, we utilize ROBL to keep the algorithm from falling into local optima. Thus, the computational complexities of position updating of RLTLBO is $O(2 \times N \times D)$, where D is the dimension size of the problem. Therefore, the total computational complexity of the proposed RLTLBO algorithm is $O(3 \times N + 2 \times N \times D)$.

4. Numerical Experiments and Results

In this section, two different kinds of benchmark functions are performed to evaluate the performance of the proposed RLTLBO algorithm. Standard benchmark functions are first tested to assess the algorithm in solving twenty-three simple numerical problems. Then, the CEC2017 benchmark functions are utilized to evaluate the algorithm in solving complex numerical problems. The RLTLBO is compared with three types of existing algorithms, including the classic methods, TLBO and SSA, the recently proposed algorithms, HOA [31], AO, and HHO [30], and the improved algorithms, mGWO [32], MALO [33] and DSCA [34]. For the consistency of all tests, we set the population size to $N = 30$, the dimension size to $D = 30$, and the maximum number of iterations to $T = 500$. All algorithms are run 30 times independently, and the average values and standard deviations are presented as the final experimental results. All experiments are implemented in MATLAB R2020b on a PC with Intel (R) Core (TM) i5-9500 CPU @ 3.00 GHz and RAM 16 GB memory on OS windows 10.

4.1. Standard Benchmark Function Experiments. Standard benchmark functions [47] can be divided into three types: unimodal, multimodal and fixed-dimension multimodal functions. Unimodal functions only have one global optimum and no local optima, which can be used to evaluate an algorithm's convergence rate and exploitation capability. Multimodal and fixed-dimension multimodal functions have a global optimum and multiple local optima. This characteristic makes these functions effective for testing the exploration and local optima avoidance abilities of an algorithm. The benchmark function details are listed in Tables 1–3.

		Action	
		Learning mode 1	Learning mode 2
State	Learning mode 1	-1	+1
	Learning mode 2	+1	-1

(a)

		Action	
		Learning mode 1	Learning mode 2
State	Learning mode 1	1.3	-0.6
	Learning mode 2	3.5	2.9

(b)

FIGURE 5: The reward table and Q-Table example of RLTLBO. (a) Reward Table sample (b) Q-Table sample.

```

(1) Set the initial values of the population size N and the maximum number of iterations T
(2) Initialize positions of the population X
(3) Initialize the reward table and Q-Table
(4) While  $t < T$ 
(5) Update Xteacher and Mean
(6) Update individual using equation (1) % teacher phase
(7) If  $\text{rand} < 0.5\%$  learner phase
(8)   If (learning mode 1 value > learning mode 2 value) % learning mode values is obtained by Q-Table
(9)     Calculate new fitness value using equation (2) % learning mode 1
(10)    If (new fitness value < current fitness value)
(11)      Reward = 1
(12)    Else
(13)      Reward = -1
(14)    End
(15)  Else
(16)    Calculate new fitness value using equation (4) % learning mode 2
(17)    If (new fitness value < current fitness value)
(18)      Reward = 1
(19)    Else
(20)      Reward = -1
(21)    End
(22)  End
(23) Else
(24)   If (learning mode 1 value > learning mode 2 value) % learning mode values is obtained by Q-Table
(25)     Calculate new fitness value using equation (2) % learning mode 1
(26)     If (new fitness value < current fitness value)
(27)       Reward = 1
(28)     Else
(29)       Reward = -1
(30)     End
(31)   Else
(32)     Calculate new fitness value using equation (4) % learning mode 2
(33)     If (new fitness value < current fitness value)
(34)       Reward = 1
(35)     Else
(36)       Reward = -1
(37)     End
(38)   End
(39) End
(40) Update Q-Table using equation (5)
(41) ROBL strategy using equation (3)
(42) End while
(43) Return Xteacher

```

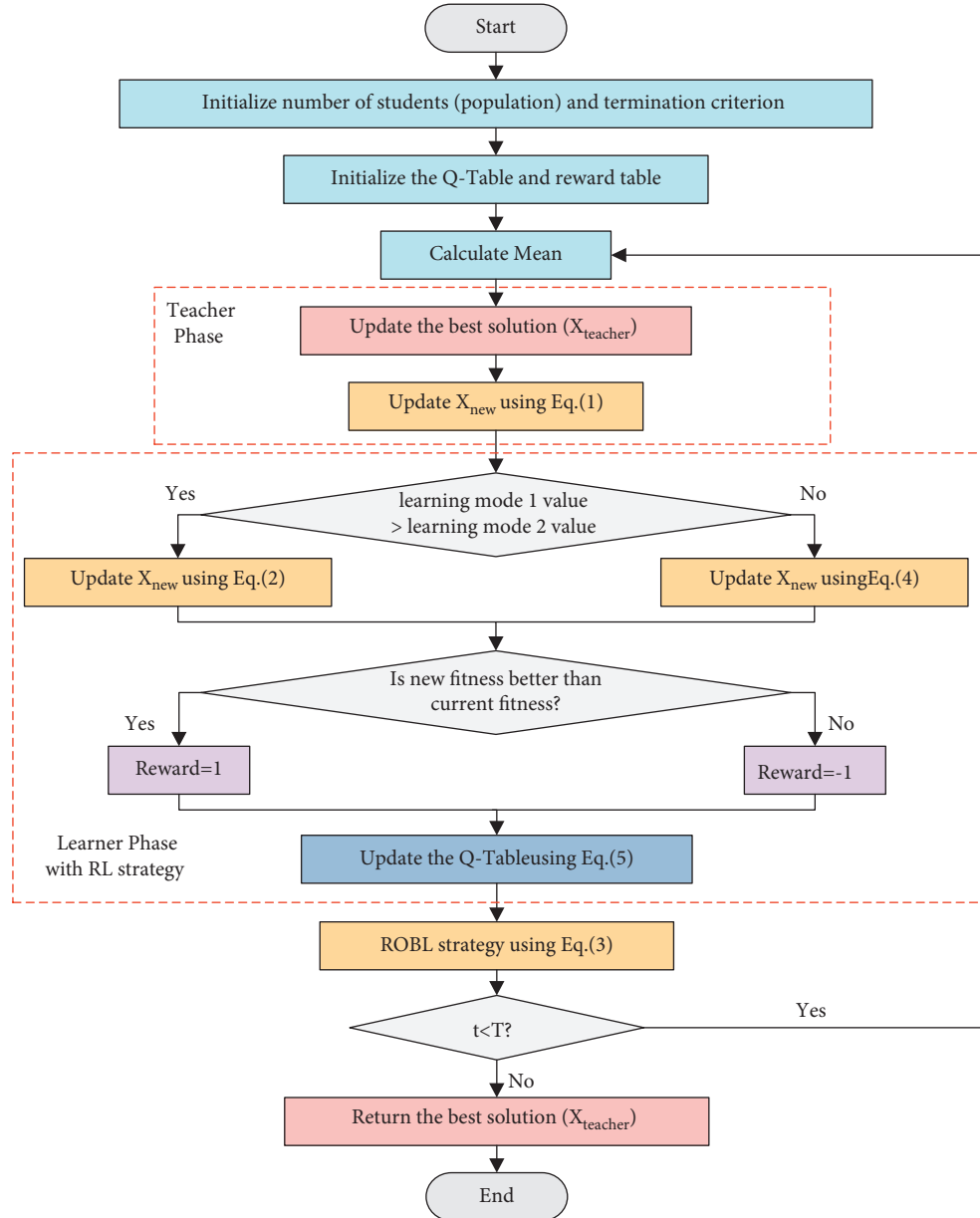


FIGURE 6: The flowchart of RLTLBO.

TABLE 1: Unimodal benchmark functions.

Function	Dim	Range	f_{\min}
$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100, 100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10, 10]	0
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	[-100, 100]	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	[-100, 100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30, 30]	0
$F_6(x) = \sum_{i=1}^n (x_i + 5)^2$	30	[-100, 100]	0
$F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	30	[-1.28, 1.28]	0

4.1.1. *Qualitative Results.* The data results of the 23 standard benchmark functions are shown in Table 4, and the optimal results are bolded. For the unimodal functions F1–F7, the RLTLBO algorithm achieves the best results among all

comparative algorithms on most functions in average values and standard deviations, and only obtains worse results on F5–F6. The RLTLBO obtains the theoretical optimum of F1 and F3. It can be concluded from the comparison results that

TABLE 2: Multimodal benchmark functions.

Function	Dim	Range	f_{\min}
$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500, 500]	$-418.9829 \times \text{Dim}$
$F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	0
$F_{10}(x) = -20 \exp(-0.2\sqrt{1/n \sum_{i=1}^n x_i^2}) - \exp(1/n \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	[-32, 32]	0
$F_{11}(x) = 1/4000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	30	[-600, 600]	0
$F_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4)$, where $y_i = 1 + x_i + 1/4$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50, 50]	0
$F_{13}(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)])$ $+ \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	[-50, 50]	0

TABLE 3: Fixed-dimension multimodal benchmark functions.

Function	Dim	Range	f_{\min}
$F_{14}(x) = (1/500 + \sum_{j=1}^{25} 1/j + \sum_{i=1}^2 (x_i - a_{ij})^6)^{-1}$	2	[-65, 65]	0.998
$F_{15}(x) = \sum_{i=1}^{11} [a_i - x_1 (b_i^2 + b_i x_2)] / b_i^2 + b_i x_3 + x_4]^2$	4	[-5, 5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1 x_2 - 4x_2^2 + x_2^4$	2	[-5, 5]	-1.0316
$F_{17}(x) = (x_2 - 5.1/4\pi^2 x_1^2 + 5/\pi x_1 - 6)^2 + 10(1 - 1/8\pi)\cos x_1 + 10$	2	[-5, 5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_2 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	[-2, 2]	3
$F_{19}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2)$	3	[-1, 2]	-3.86
$F_{20}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2)$	6	[0, 1]	-3.32
$F_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.1532
$F_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

TABLE 4: Results of algorithms on 23 standard benchmark functions.

Function	RLTLBO	TLBO	mGWO	MALO	DSCA	HOA	AO	HHO	SSA	
F1	Mean	0.00E+00	3.90E-79	4.26E-19	1.37E-03	2.55E-288	3.13E-136	2.34E-104	8.97E-98	1.30E-07
	Std	0.00E+00	6.59E-79	1.08E-18	1.56E-03	0.00E+00	1.21E-135	1.08E-103	4.16E-97	1.09E-07
F2	Mean	1.29E-223	4.17E-40	3.37E-12	6.86E+01	5.92E-171	4.44E-68	2.82E-53	1.34E-48	1.79E+00
	Std	0.00E+00	3.21E-40	2.54E-12	4.90E+01	0.00E+00	2.42E-67	1.13E-52	5.75E-48	1.15E+00
F3	Mean	0.00E+00	2.50E-17	6.41E-01	4.81E+03	1.43E-241	2.23E+02	2.22E-101	7.16E-79	1.61E+03
	Std	0.00E+00	4.35E-17	1.46E+00	2.18E+03	0.00E+00	5.03E+02	1.22E-100	3.56E-78	1.03E+03
F4	Mean	3.07E-221	1.72E-32	2.42E-03	1.64E+01	1.97E-134	5.04E-65	3.20E-53	2.51E-48	1.11E+01
	Std	0.00E+00	1.76E-32	3.02E-03	4.23E+00	1.08E-133	1.84E-64	1.75E-52	8.46E-48	3.74E+00
F5	Mean	2.65E+01	2.42E+01	2.64E+01	9.86E-01	2.85E+01	2.89E+01	6.82E-03	1.22E-02	2.55E+02
	Std	4.01E-01	7.41E-01	8.44E-01	5.21E+00	3.59E-01	7.45E-02	1.66E-02	1.79E-02	3.44E+02
F6	Mean	9.03E-02	2.57E-06	4.54E-01	5.00E-04	6.01E+00	6.46E+00	4.43E-05	9.58E-05	1.28E-07
	Std	1.15E-01	7.98E-06	3.20E-01	3.05E-04	1.61E-01	4.76E-01	6.15E-05	1.24E-04	1.13E-07
F7	Mean	3.57E-05	1.12E-03	4.61E-03	1.05E-04	2.54E-04	5.88E-02	9.62E-05	1.68E-04	1.81E-01
	Std	4.71E-05	3.06E-04	1.64E-03	7.89E-05	2.88E-04	4.10E-02	7.92E-05	1.36E-04	8.96E-02
F8	Mean	-7.36E+03	-7.85E+03	-6.58E+03	-1.22E+04	-3.96E+03	-4.30E+03	-8.92E+03	-1.25E+04	-7.56E+03
	Std	6.78E+02	9.32E+02	1.24E+03	1.08E+03	4.31E+02	7.82E+02	3.77E+03	8.42E+01	7.07E+02

TABLE 4: Continued.

Function		RLTLBO	TLBO	mGWO	MALO	DSCA	HOA	AO	HHO	SSA
F9	Mean	0.00E+00	1.41E+01	1.70E+01	8.44E+01	0.00E+00	5.06E+01	0.00E+00	0.00E+00	5.19E+01
	Std	0.00E+00	6.20E+00	9.11E+00	3.15E+01	0.00E+00	9.32E+01	0.00E+00	0.00E+00	1.88E+01
F10	Mean	8.88E-16	7.05E-15	1.14E+00	4.77E+00	8.88E-16	6.10E-15	8.88E-16	8.88E-16	2.62E+00
	Std	0.00E+00	1.60E-15	1.88E+00	2.64E+00	0.00E+00	2.42E-15	0.00E+00	0.00E+00	8.98E-01
F11	Mean	0.00E+00	3.29E-04	4.86E-03	6.05E-02	0.00E+00	1.18E-01	0.00E+00	0.00E+00	2.24E-02
	Std	0.00E+00	1.80E-03	9.13E-03	2.33E-02	0.00E+00	2.57E-01	0.00E+00	0.00E+00	1.45E-02
F12	Mean	8.32E-04	5.38E-07	3.51E-02	1.60E-05	8.37E-01	1.23E+00	3.04E-06	1.02E-05	7.22E+00
	Std	1.52E-03	2.76E-06	4.56E-02	1.16E-05	1.08E-01	2.42E-01	4.59E-06	1.12E-05	3.01E+00
F13	Mean	2.00E+00	7.41E-02	3.83E-01	1.70E-03	2.76E+00	3.08E+00	4.57E-05	8.69E-05	2.19E+01
	Std	1.17E+00	8.70E-02	2.15E-01	3.95E-03	5.11E-02	1.83E-01	1.18E-04	9.70E-05	1.44E+01
F14	Mean	1.06E+00	9.98E-01	9.98E-01	1.46E+00	1.35E+00	2.78E+00	4.06E+00	1.36E+00	1.16E+00
	Std	3.62E-01	0.00E+00	3.81E-12	7.69E-01	6.1E-01	2.07E+00	4.46E+00	9.52E-01	4.57E-01
F15	Mean	3.55E-04	3.82E-04	3.04E-03	1.40E-03	8.91E-04	6.77E-03	5.00E-04	4.01E-04	3.55E-03
	Std	1.02E-04	1.54E-04	6.91E-03	3.62E-03	3.99E-04	5.47E-03	1.10E-04	2.36E-04	6.71E-03
F16	Mean	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-9.99E-01	-1.03E+00	-1.03E+00	-1.03E+00
	Std	6.58E-16	6.95E-16	3.39E-08	1.65E-13	3.99E-04	3.29E-02	3.01E-04	3.76E-09	1.83E-14
F17	Mean	3.98E-01	3.98E-01	3.98E-01	3.98E-01	4.09E-01	3.99E-01	3.98E-01	3.98E-01	3.98E-01
	Std	0.00E+00	0.00E+00	6.52E-09	5.57E-14	1.06E-02	1.08E-03	1.09E-04	4.60E-06	7.21E-15
F18	Mean	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	4.94E+00	3.03E+00	3.00E+00	3.00E+00
	Std	4.95E-16	1.24E-15	1.03E-07	5.76E-13	8.33E-04	6.82E+00	5.73E-02	3.88E-07	2.87E-13
F19	Mean	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.82E+00	-3.86E+00	-3.85E+00	-3.86E+00	-3.86E+00
	Std	2.71E-15	3.16E-15	1.08E-06	6.39E-13	2.33E-02	6.99E-04	6.96E-03	2.07E-03	1.09E-12
F20	Mean	-3.31E+00	-3.30E+00	-3.23E+00	-3.23E+00	-2.80E+00	-3.25E+00	-3.16E+00	-3.08E+00	-3.23E+00
	Std	2.95E-02	4.12E-02	6.47E-02	5.14E-02	2.71E-01	9.05E-02	8.91E-02	1.22E-01	6.22E-02
F21	Mean	-1.02E+01	-1.02E+01	-9.98E+00	-7.62E+00	-3.27E+00	-9.43E+00	-1.01E+01	-5.18E+00	-8.07E+00
	Std	6.04E-09	1.41E-03	9.30E-01	2.82E+00	1.54E+00	9.62E-01	2.09E-02	7.51E-01	3.28E+00
F22	Mean	-1.04E+01	-1.01E+01	-1.04E+01	-7.06E+00	-3.87E+00	-9.36E+00	-1.04E+01	-5.08E+00	-9.32E+00
	Std	1.23E-07	1.25E+00	4.45E-04	3.48E+00	1.17E+00	1.69E+00	5.50E-02	6.94E-03	2.51E+00
F23	Mean	-1.05E+01	-1.01E+01	-1.05E+01	-7.31E+00	-4.19E+00	-9.63E+00	-1.05E+01	-5.24E+00	-7.89E+00
	Std	1.57E-07	1.57E+00	3.42E-04	3.55E+00	1.11E+00	1.52E+00	2.23E-02	9.58E-01	3.59E+00

RLTLBO has strong competitiveness in the unimodal functions, which indicates that the excellent exploitation capability comes from the RL mechanism.

For the multimodal and fixed-dimension multimodal functions F8–F23, it can be seen from Table 4 that RLTLBO achieves the smallest average values and standard deviations on 12 of all 16 test functions compared to other methods, which indicates a very high accuracy and stability. Several poor results appear on F8 and F12–F14, but they are not the worst results. The satisfying results on the multimodal and fixed-dimension multimodal functions prove that the exploration and local optima avoidance capabilities of the RLTLBO are excellent, which might be derived from the ROBL strategy.

Figure 7 provides the convergence curves of RLTLBO and the comparative algorithms for 23 standard benchmark functions. The convergence rate reflected by convergence curves can show us the improvement of exploration and exploitation more intuitively. For F1–F4, F7, F9–F11, and F15–F21, the RLTLBO presents a faster convergence speed than other meta-heuristic algorithms, and the convergence accuracy is also the best. The RLTLBO is ranked in the second position in terms of convergence speed for F22 and F23. For benchmark functions F5–F6, F8, and F12–F14, the RLTLBO does not perform very well, the same as the results in Table 4.

4.1.2. The Wilcoxon Test. The Wilcoxon rank-sum test [48] results are listed in Table 5, which can assess the statistical performance differences between the RLTLBO algorithm and the comparative algorithms. A p -value less than 0.05 indicates a substantial difference between the two compared methods. It is obvious that the overwhelming majority p -values in Table 5 are less than 0.05, indicating that there are statistically and substantial differences between RLTLBO and the other methods. Combining the results in Table 4, it can be concluded that the RLTLBO algorithm outperforms the others. The competitive results of RLTLBO indicate that this algorithm has high capabilities of exploration and exploitation. In summary, the RLTLBO algorithm provides better results than other comparative algorithms.

4.2. CEC2017 Benchmark Function Experiments. Standard benchmark function experiments prove the superior performance on simple optimization problems of the proposed RLTLBO algorithm. CEC2017 [49], one of the most challenging test suites, can help check the performance of complex optimization problems. Some hybrid and composition functions are selected to further test the performance of RLTLBO. These types of functions are precisely what the standard test functions do not have. The functional details and the comparison results are presented in Tables 6 and 7. As mentioned above, each method runs 30 times with

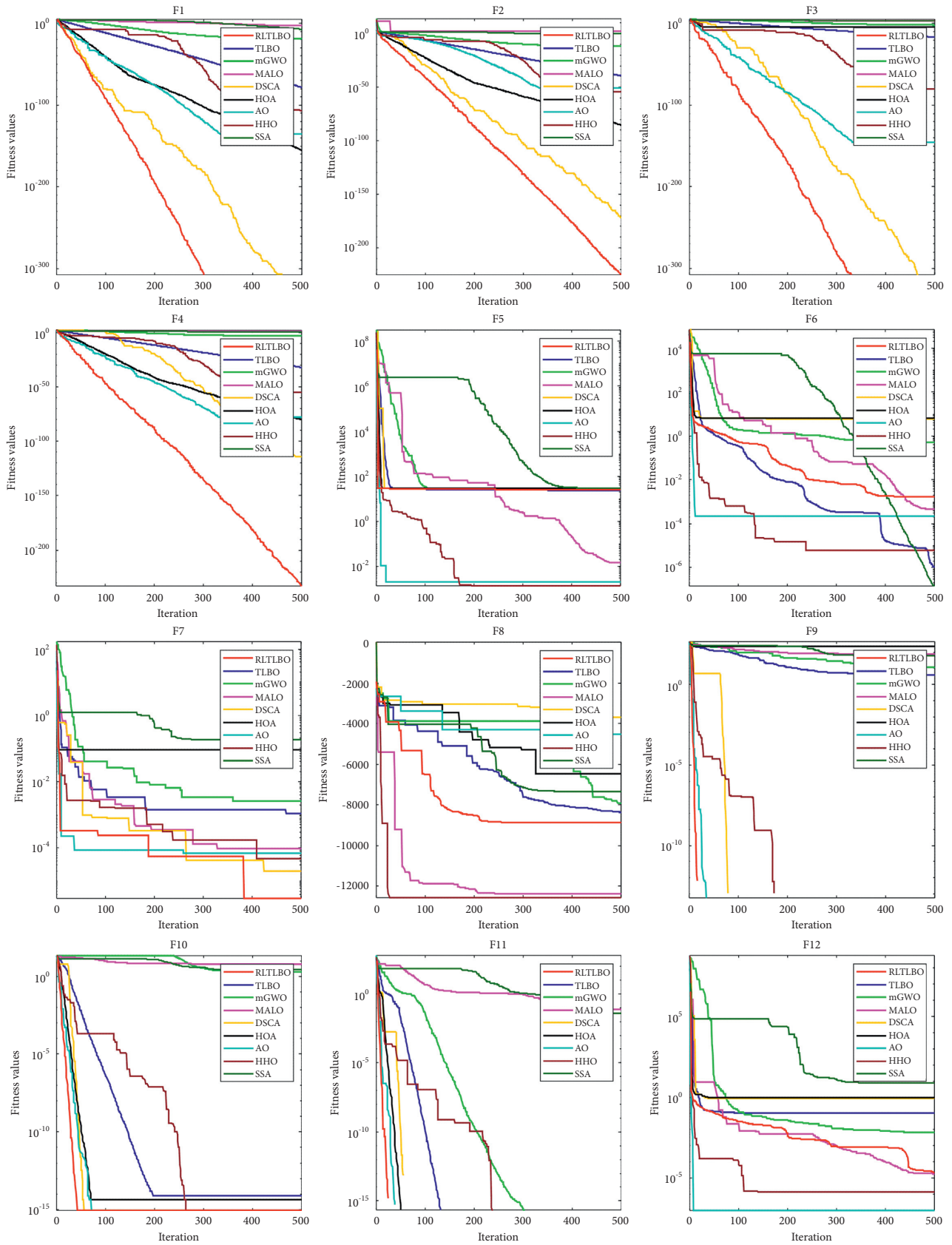


Figure 7: Continued.

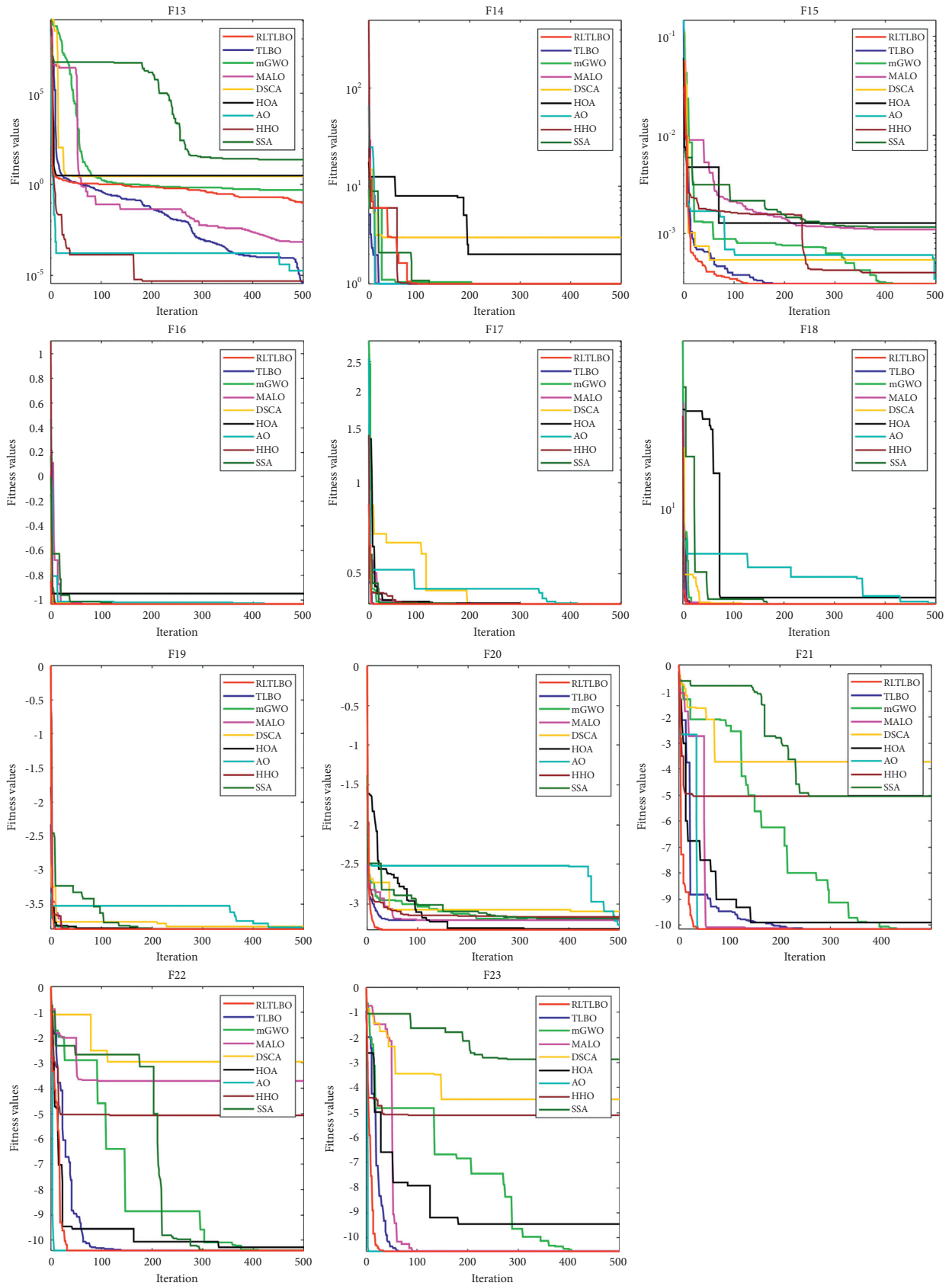


FIGURE 7: Convergence curves of 23 standard benchmark functions.

TABLE 5: p -Values from the Wilcoxon rank-sum test for the results in Table 4.

Function	RLTLBO vs.							
	TLBO	mGWO	MALO	DSCA	HOA	AO	HHO	SSA
F1	6.10E-05	6.10E-05	6.10E-05	NaN	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F2	6.10E-05	6.10E-05	6.10E-05	6.10E-04	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F3	6.10E-05	6.10E-05	6.10E-05	1.56E-02	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F4	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F5	6.10E-05	3.30E-01	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	8.54E-04
F6	6.10E-05	1.22E-04	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F7	6.10E-05	6.10E-05	4.89E-01	6.10E-04	6.10E-05	4.89E-01	7.30E-02	6.10E-05
F8	0.010254	6.37E-02	6.10E-05	6.10E-05	6.10E-05	1.21E-01	6.10E-05	5.61E-01
F9	6.10E-05	6.10E-05	6.10E-05	NaN	1.25E-01	NaN	NaN	6.10E-05
F10	6.10E-05	6.10E-05	6.10E-05	NaN	6.10E-05	NaN	NaN	6.10E-05
F11	NaN	1.95E-03	6.10E-05	NaN	3.12E-02	NaN	NaN	6.10E-05
F12	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F13	3.05E-04	6.10E-04	6.10E-05	3.89E-01	2.01E-03	6.10E-05	6.10E-05	3.05E-04
F14	NaN	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F15	8.90E-01	2.01E-03	1.83E-04	6.10E-05	6.10E-05	6.10E-05	8.36E-03	6.10E-05
F16	NaN	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	1.22E-04	6.10E-05
F17	NaN	6.10E-05	2.44E-04	6.10E-05	6.10E-05	6.10E-05	6.10E-05	9.76E-04
F18	NaN	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F19	NaN	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05
F20	8.52E-01	4.13E-02	1.35E-01	6.10E-05	2.01E-03	6.10E-05	6.10E-05	3.05E-04
F21	1.68E-01	6.10E-05	4.79E-02	6.10E-05	6.10E-05	6.10E-05	6.10E-05	1.03E-02
F22	6.25E-02	6.10E-05	2.56E-02	6.10E-05	6.10E-05	6.10E-05	6.10E-05	4.13E-02
F23	7.81E-03	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	6.10E-05	2.56E-02

TABLE 6: Descriptions of the benchmark functions from CEC2017.

Function	Name	Dim	Range	f_{min}
Hybrid functions (N is basic number of functions)				
C13	Hybrid function 3 ($N=3$)	10	[-100, 100]	1300
C14	Hybrid function 4 ($N=4$)	10	[-100, 100]	1400
C15	Hybrid function 5 ($N=4$)	10	[-100, 100]	1500
C19	Hybrid function 6 ($N=5$)	10	[-100, 100]	1900
Composite functions (N is basic number of functions)				
C22	Composite function 2 ($N=3$)	10	[-100, 100]	2200
C25	Composite function 5 ($N=5$)	10	[-100, 100]	2500
C28	Composite function 8 ($N=6$)	10	[-100, 100]	2800
C29	Composite function 9 ($N=6$)	10	[-100, 100]	2900

TABLE 7: Comparison results of algorithms on CEC2017.

Function		RLTLBO	TLBO	mGWO	MALO	DSCA	HOA	AO	HHO	SSA
C13	Mean	4.38E+03	6.04E+03	4.35E+03	1.78E+04	6.25E+05	1.53E+06	1.77E+04	1.70E+04	1.46E+04
	Std	2.76E+03	4.33E+03	2.99E+03	1.30E+04	4.55E+05	1.28E+06	1.39E+04	1.03E+04	1.29E+04
C14	Mean	1.46E+03	1.47E+03	1.47E+03	2.75E+03	4.78E+03	3.87E+03	2.36E+03	2.20E+03	3.35E+03
	Std	1.81E+01	2.40E+01	1.98E+01	2.02E+03	3.76E+03	1.99E+03	1.12E+03	1.05E+03	3.10E+03
C15	Mean	1.62E+03	1.73E+03	1.74E+03	8.28E+03	7.97E+03	2.49E+04	5.91E+03	7.35E+03	1.06E+04
	Std	5.96E+01	1.44E+02	2.36E+02	5.72E+03	3.62E+03	1.54E+04	2.16E+03	3.10E+03	7.51E+03
C19	Mean	2.00E+03	2.11E+03	2.65E+03	1.54E+04	3.37E+04	1.69E+04	2.10E+04	1.67E+04	8.46E+03
	Std	9.63E+00	3.19E+02	1.68E+03	1.23E+04	3.00E+04	1.34E+04	2.88E+04	1.37E+04	6.44E+03
C22	Mean	2.30E+03	2.30E+03	2.30E+00	2.30E+03	2.55E+03	2.47E+03	2.31E+03	2.41E+03	2.33E+03
	Std	1.99E+01	8.68E+00	9.25E-01	2.88E+01	8.10E+01	4.58E+02	5.85E+00	3.85E+02	1.69E+02
C25	Mean	2.92E+03	2.93E+03	2.92E+03	2.93E+03	3.12E+03	2.97E+03	2.94E+03	2.93E+03	2.92E+03
	Std	2.32E+01	2.41E+01	2.33E+01	2.38E+01	6.48E+01	2.35E+01	2.50E+01	6.24E+01	2.45E+01
C28	Mean	3.23E+03	3.30E+03	3.33E+03	3.31E+03	3.40E+03	3.50E+03	3.44E+03	3.45E+03	3.29E+03
	Std	1.15E+02	1.60E+02	1.12E+02	1.47E+02	9.48E+01	1.06E+02	1.09E+02	1.45E+02	1.68E+02
C29	Mean	3.18E+03	3.19E+03	3.17E+03	3.27E+03	3.38E+03	3.38E+03	3.26E+03	3.37E+03	3.27E+03
	Std	1.84E+01	2.16E+01	2.13E+01	6.15E+01	5.77E+01	6.58E+01	5.87E+01	1.20E+02	7.20E+01

30 search agents and 500 iterations. From Table 7, the proposed RLTLBO achieves both the best average and standard deviation values on five of the eight all functions. For the remaining three functions, RLTLBO obtains one of the best average and standard deviation values. The RLTLBO completely exceeds the TLBO, MALO, HOA, AO, HHO, and SSA methods completely. The statistical results are also listed in Table 8. There are only seven p -values greater than 0.05 in all test functions, which means considerable differences between the RLTLBO and the compared methods. These results suggest that RLTLBO can achieve great results on complex problems as well.

5. Experiments on Industrial Engineering Design Problems

In this section, eight well-known constrained industrial engineering design problems, including the welded beam design problem, pressure vessel design problem, tension and compression spring design problem, speed reducer design problem, three-bar truss design problem, car crashworthiness design problem, tubular column design problem, and frequency-modulated sound wave design problem, are solved to further verify the performance of the proposed RLTLBO algorithm. The results of RLTLBO are compared to various optimization methods proposed in previous studies.

5.1. Welded Beam Design Problem. The purpose of this problem is to minimize the cost of the welded beam (Figure 8). Four variables need to be optimized: the thickness of weld (h), the thickness of the bar (b), length of the bar (l), and height of the bar (t). The mathematical formulation is listed as follows:

Consider $\vec{z} = [z_1, z_2, z_3, z_4] = [h, l, t, b]$.

Minimize $f(\vec{z}) = 1.10471z_1^2z_2 + 0.04811z_3z_4 (14.0 + z_2)$, subject to

$$\begin{cases} g_1(\vec{z}) = \tau(\vec{z}) - \tau_{\max} \leq 0, \\ g_2(\vec{z}) = \sigma(\vec{z}) - \sigma_{\max} \leq 0, \\ g_3(\vec{z}) = \delta(\vec{z}) - \delta_{\max} \leq 0, \\ g_4(\vec{z}) = z_1 - z_4 \leq 0, \\ g_5(\vec{z}) = P - P_c(\vec{z}) \leq 0, \\ g_6(\vec{z}) = 0.125 - z_1 \leq 0, \\ g_7(\vec{z}) = 1.10471z_1^2 + 0.04811z_3z_4(14.0 + z_2) - 5.0 \leq 0. \end{cases} \quad (6)$$

Variable range

$$\begin{cases} 0.1 \leq z_1 \leq 2, \\ 0.1 \leq z_2 \leq 10, \\ 0.1 \leq z_3 \leq 10, \\ 0.1 \leq z_4 \leq 2, \end{cases} \quad (7)$$

where

$$\begin{cases} \tau(\vec{z}) = \sqrt{\tau'^2 + 2\tau'\tau''\frac{z_2}{2R} + \tau''^2}, \\ \tau' = \frac{P}{\sqrt{2}z_1z_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{z_2}{2}\right), \\ R = \sqrt{\frac{z_2^2}{4} + \left(\frac{z_1 + z_3}{2}\right)^2}, \\ J = 2\left\{\sqrt{2}z_1z_2\left[\frac{z_2^2}{12} + \left(\frac{z_1 + z_3}{2}\right)^2\right]\right\}, \\ \sigma(\vec{z}) = \frac{6PL}{z_4z_3^2}, \delta(\vec{z}) = \frac{4PL^3}{Ez_3^3z_4}, \\ P_c(\vec{z}) = \frac{4.013E\sqrt{z_3^2z_4^6/36}}{L^2}\left(1 - \frac{z_3}{2L}\sqrt{\frac{E}{4G}}\right), \\ P = 6000lb, L = 14in, \delta_{\max} = 0.25in, \\ E = 30 \times 10^6 psi, G = 12 \times 10^6 psi, \\ \tau_{\max} = 13600 psi, \sigma_{\max} = 30000 psi. \end{cases} \quad (8)$$

The RLTLBO is compared to SMA [50], WOA, MPA [51], MVO [52], GA, and HS [53] methods. The comparison results presented in Table 9 show the superior of the RLTLBO algorithm with a smaller cost than other algorithms.

5.2. Pressure Vessel Design Problem. The objective of this problem is to minimize the fabrication cost of the cylindrical pressure vessel to meet the pressure requirements. As shown in Figure 9, four structural parameters in this problem need to be minimized, including the thickness of the shell (T_s), the thickness of the head (T_h), inner radius (R), and the length of the cylindrical section without the head (L). The formulation of four optimization constraints can be described as follows:

Consider $\vec{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]$.

Minimize $f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$, subject to

$$\begin{cases} g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0, \\ g_2(\vec{x}) = -x_3 + 0.00954x_3 \leq 0, \\ g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0, \\ g_4(\vec{x}) = x_4 - 240 \leq 0. \end{cases} \quad (9)$$

TABLE 8: p values from the Wilcoxon rank-sum test for the results in Table 7.

Function	RLTLBO vs.							
	TLBO	mGWO	MALO	DSCA	HOA	AO	HHO	SSA
C13	$2.90E-02$	$3.59E-01$	$1.81E-02$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$3.36E-03$	$1.22E-04$
C14	$1.35E-01$	$4.37E-02$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$1.83E-04$	$6.10E-05$
C15	$3.36E-03$	$8.36E-03$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$6.10E-05$
C19	$1.24E-02$	$3.30E-02$	$8.54E-04$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$6.10E-05$
C22	$4.27E-03$	$4.13E-02$	$4.04E-02$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$8.47E-02$
C25	$5.61E-01$	$8.47E-01$	$8.47E-01$	$6.10E-05$	$2.01E-03$	$1.21E-02$	$1.69E-02$	$3.62E-01$
C28	$2.48E-02$	$1.51E-02$	$4.79E-02$	$1.81E-02$	$1.53E-03$	$6.10E-04$	$5.37E-03$	$4.21E-02$
C29	$4.54E-03$	$6.10E-04$	$6.10E-05$	$6.10E-05$	$6.10E-05$	$8.54E-04$	$6.10E-05$	$1.51E-02$

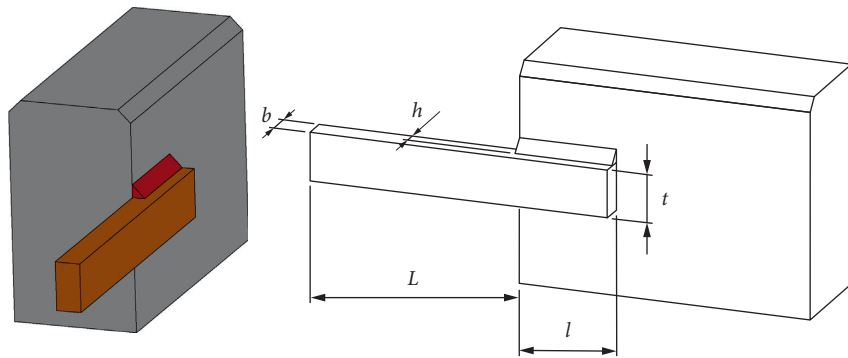


FIGURE 8: Welded beam design problem.

TABLE 9: Comparison results for the welded beam design problem.

Algorithm	Optimum variables				Optimum cost
	h	l	t	b	
RLTLBO	0.205730	3.253000	9.036600	0.205730	1.695200
SMA [50]	0.205400	3.258900	9.038400	0.205800	1.696040
WOA [14]	0.205396	3.484293	9.037426	0.206276	1.730499
MPA [51]	0.205728	3.470509	9.036624	0.205730	1.724853
MVO [52]	0.205463	3.473193	9.044502	0.205695	1.726450
GA [6]	0.248900	6.173000	8.178900	0.253300	2.430000
HS [53]	0.244200	6.223100	8.291500	0.240000	2.380700

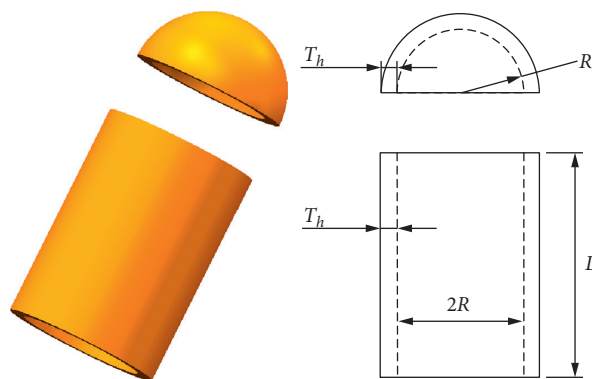


FIGURE 9: Pressure vessel design problem.

Variable range

$$\begin{cases} 0 \leq x_1 \leq 99, \\ 0 \leq x_2 \leq 99, \\ 10 \leq x_3 \leq 200, \\ 10 \leq x_4 \leq 200. \end{cases} \quad (10)$$

From the results in Table 10, it is obvious that RLTLBO can obtain superior optimal values compared to AO, SMA, WOA, GWO, MVO, GA, and ES [54].

5.3. Tension/Compression Spring Design Problem. This problem aims to minimize the weight of the tension/compression spring (Figure 10). Three variables need to be optimized, including the wire diameter (d), the number of active coils (N), and mean coil diameter (D). This problem can be described as follows:

Consider $\vec{x} = [x_1 \ x_2 \ x_3] = [d \ D \ N]$.

Minimize $f(\vec{x}) = (x_3 + 2)x_2x_1^2$, subject to

$$\begin{cases} g_1(\vec{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0, \\ g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0, \\ g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\ g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0. \end{cases} \quad (11)$$

Variable range

$$\begin{cases} 0.05 \leq x_1 \leq 2.00, \\ 0.25 \leq x_2 \leq 1.30, \\ 2.00 \leq x_3 \leq 15.00. \end{cases} \quad (12)$$

The RLTLBO is compared to AO, SSA, WOA, GWO, PSO, GA, and HS algorithms. Results are listed in Table 11 and show that the RLTLBO can obtain the best weight compared to all other algorithms.

5.4. Speed Reducer Design Problem. In this case, the purpose is to minimize the weight of the speed reducer (Figure 11). Seven variables are considered, including face width (x_1), a module of teeth (x_2), a discrete design variable on behalf of the teeth in the pinion (x_3), length of the first shaft between bearings (x_4), length of the second shaft between bearings (x_5), diameters of the first shaft (x_6), and diameters of the second shaft (x_7). The mathematical formulation is listed as follows:

Minimize

$$\begin{aligned} f(\vec{x}) = & 0.7854x_1x_2^2 + 3.3333x_3^2 + 14.9334x_3 - 43.0934 \\ & - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3), \end{aligned} \quad (13)$$

subject to

$$\begin{cases} g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0, \\ g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0, \\ g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0, \\ g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0, \\ g_5(\vec{x}) = \frac{\sqrt{(745x_4/x_2x_3)^2 + 16.9 \times 10^6}}{110.0x_6^3} - 1 \leq 0, \\ g_6(\vec{x}) = \frac{\sqrt{(745x_4/x_2x_3)^2 + 157.5 \times 10^6}}{85.0x_6^3} - 1 \leq 0, \\ g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \leq 0, \\ g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \leq 0, \\ g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \leq 0, \\ g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0, \\ g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0. \end{cases} \quad (14)$$

Variable range

$$\begin{cases} 2.6 \leq x_1 \leq 3.6, \\ 0.7 \leq x_2 \leq 0.8, \\ 17 \leq x_3 \leq 28, \\ 7.3 \leq x_4 \leq 8.3, \\ 7.8 \leq x_5 \leq 8.3, \\ 2.9 \leq x_6 \leq 3.9, \\ 5.0 \leq x_7 \leq 5.5. \end{cases} \quad (15)$$

TABLE 10: Comparison results for the pressure vessel design problem.

Algorithm	Optimum variables				Optimum cost
	Ts	Th	R	L	
RTLBO	0.7698901	0.4201098	42.536830	171.348900	5926.77920
AO [15]	1.0540000	0.1828060	59.621900	38.8050000	5949.22580
SMA [50]	0.7931000	0.3932000	40.671100	196.217800	5994.18570
WOA [14]	0.8125000	0.4375000	42.098270	176.638998	6059.74100
GWO [13]	0.8125000	0.4345000	42.089200	176.758700	6051.56390
MVO [52]	0.8125000	0.4375000	42.090738	176.738690	6060.80660
GA [6]	0.8125000	0.4375000	42.097398	176.654050	6059.94634
ES [54]	0.8125000	0.4375000	42.098087	176.640518	6059.74560

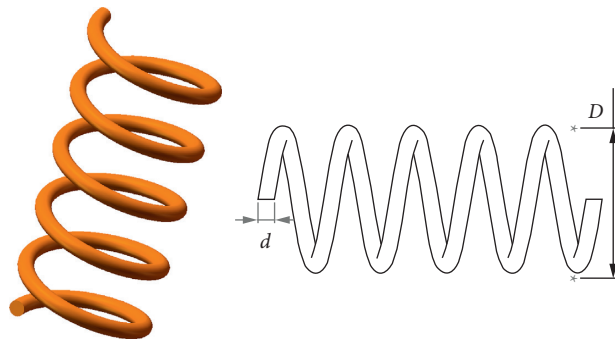


FIGURE 10: Tension/compression spring design problem.

TABLE 11: Comparison results for the tension/compression spring design problem.

Algorithm	Optimum variables			Optimum weight
	d	D	N	
RTLBO	0.0551180	0.505900	5.1167000	0.01093800
AO [15]	0.0502439	0.352620	10.542500	0.01116500
SSA [12]	0.0512070	0.345215	12.004032	0.01267630
WOA [14]	0.0512070	0.345215	12.004032	0.01267630
GWO [13]	0.0516900	0.356737	11.288850	0.01266600
PSO [11]	0.0517280	0.357644	11.244543	0.01267470
GA [6]	0.0514800	0.351661	11.632201	0.01270478
HS [53]	0.0511540	0.349871	12.076432	0.01267060

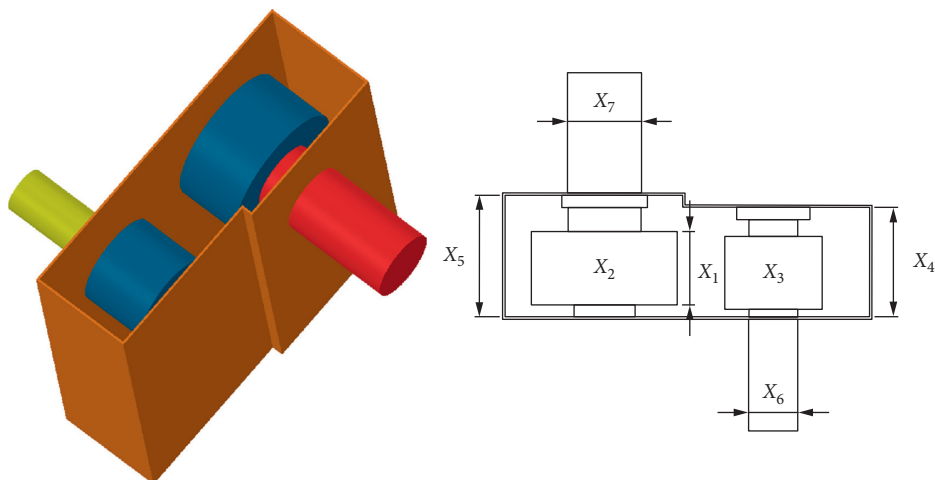


FIGURE 11: Speed reducer design problem.

Compared to AO, PSO, AOA, GA, SCA [55], HS, and FA [56], RLTLBO achieves better results in the speed reducer problem, as shown in Table 12.

5.5. Three-Bar Truss Design Problem. The three-bar truss design problem aims to minimize the weight of a truss with three bars by controlling the length of three bars (A1, A2, and A3) (Figure 12). Three main constraints need to be satisfied, including deflection, stress, and buckling. The mathematical form of this problem is given:

Consider $\vec{x} = [x_1 \ x_2] = [A_1 \ A_2]$.

Minimize $f(\vec{x}) = (2\sqrt{2}x_1 + x_2) * l$ subject to

$$\begin{cases} g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0, \\ g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0, \\ g_3(\vec{x}) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0. \end{cases} \quad (16)$$

Consider $0 \leq x_1, x_2 \leq 1$, where $l = 100\text{cm}$, $P = 2\text{KN/cm}^2$, $\sigma = 2\text{KN/cm}^2$.

The result of RLTLBO is listed in Table 13, compared to AO, SSA, AOA, MVO, and GOA [57]. It can be observed that RLTLBO outperforms other algorithms in the literature.

5.6. Car Crashworthiness Design Problem. The car crashworthiness design problem aims to minimize the weight by optimizing eleven influence variables [58], including the thickness of B-Pillar inner (x1), B-pillar reinforcement (x2), floor side inner (x3), cross members (x4), door beam (x5), door beltline reinforcement (x6) and roof rail (x7), materials of B-Pillar inner (x8) and floor side inner (x9), barrier height (x10), and barrier hitting position (x11). This problem can be formulated as follows.

Minimize

$$f(\vec{x}) = \text{weight}, \quad (17)$$

subject to

$$\begin{cases} g_1(\vec{x}) = F_a (\text{load in abdomen}) \leq 1 \text{ kN}, \\ g_2(\vec{x}) = V \times Cu (\text{dummy upper chest}) \leq 0.32 \text{ m/s}, \\ g_3(\vec{x}) = V \times Cm (\text{dummy middle chest}) \leq 0.32 \text{ m/s}, \\ g_4(\vec{x}) = V \times Cl (\text{dummy lower chest}) \leq 0.32 \text{ m/s}, \\ g_5(\vec{x}) = \Delta_{ur} (\text{upper rib deflection}) \leq 32 \text{ mm}, \\ g_6(\vec{x}) = \Delta_{mr} (\text{middle rib deflection}) \leq 32 \text{ mm}, \\ g_7(\vec{x}) = \Delta_{lr} (\text{lower rib deflection}) \leq 32 \text{ mm}, \\ g_8(\vec{x}) = F (\text{public force})_p \leq 4 \text{ kN}, \\ g_9(\vec{x}) = V_{MBP} (\text{velocity of V - pillar at middle point}) \leq 9.9 \text{ mm/ms}, \\ g_{10}(\vec{x}) = V_{FD} (\text{velocity of front door at V - pillar}) \leq 15.7 \text{ mm/ms}. \end{cases} \quad (18)$$

Variable range

$$\begin{cases} 0.5 \leq x_1 - x_7 \leq 1.5, \\ x_8, x_9 \in (0.192, 0.345), \\ -30 \leq x_{10}, x_{11} \leq 30. \end{cases} \quad (19)$$

The RLTLBO and DE, GA, FA, CS [59], GOA, and EOBL-GOA [58] are applied to solve the car crashworthiness problem. As shown in Table 14, compared to other methods, the proposed RLTLBO achieves the best result than others.

5.7. Tubular Column Design Problem. The main intention is to find a minimum cost for a uniform column, making the tubular section be able to carry a compressive load $P = 2,500$ kgf. The column is made of a material with a yield stress (σ_y) of 500 kgf/cm^2 , a modulus of elasticity (E) of $0.85 \times 10^6 \text{ kgf/cm}^2$, and a density (ρ) equal to 0.0025 kgf/cm^3 . The length (L) of the column is 250 cm . The cost of the column consists of material and construction costs. This problem is shown in Figure 13, and the optimization model of the problem is listed as follows.

Minimize $f(d, t) = 9.8dt + 2d$ subject to

TABLE 12: Comparison results for the speed reducer design problem.

Algorithm	Optimum variables							Optimum weight
	x1	x2	x3	x4	x5	x6	x7	
RTLBO	3.497600	0.7000	17.0000	7.30000	7.800000	3.350060	5.285530	2995.43740
AO [15]	3.502100	0.7000	17.0000	7.30990	7.747600	3.364100	5.299400	3007.73280
PSO [11]	3.500100	0.7000	17.0002	7.51770	7.783200	3.350800	5.286700	3145.92200
AOA [9]	3.503840	0.7000	17.0000	7.30000	7.729330	3.356490	5.286700	2997.91570
GA [6]	3.510253	0.7000	17.0000	8.35000	7.800000	3.362201	5.287723	3067.56100
SCA [55]	3.508755	0.7000	17.0000	7.30000	7.800000	3.461020	5.289213	3030.56300
HS [53]	3.520124	0.7000	17.0000	8.37000	7.800000	3.366970	5.288719	3029.00200
FA [56]	3.507495	0.7001	17.0000	7.719674	8.080854	3.351512	5.287051	3010.13749

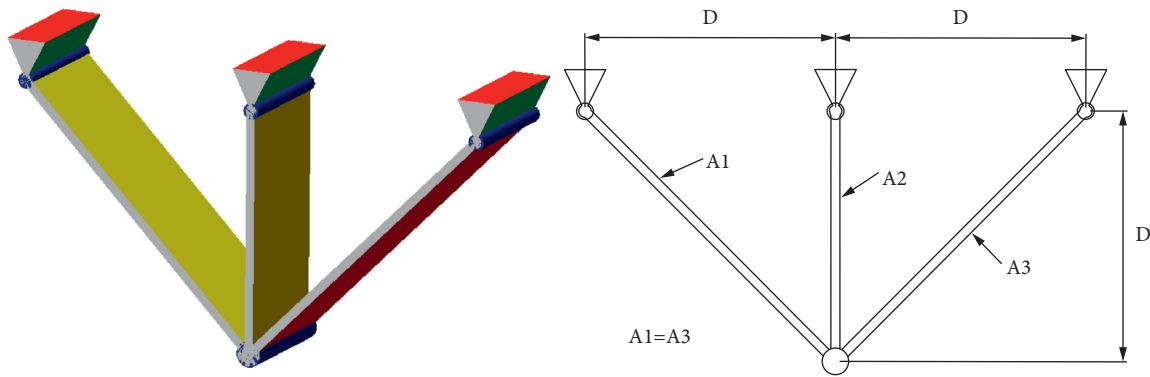


FIGURE 12: Three-bar truss design problem.

TABLE 13: Comparison results for the three-bar truss design problem.

Algorithm	Optimum variables		Optimum weight
	x1	x2	
RTLBO	0.7884200000000000	0.4081100000000000	263.8523000000000
AO [15]	0.7926000000000000	0.3966000000000000	263.8684000000000
SSA [12]	0.7886654100000000	0.4082757840000000	263.8958400000000
AOA [9]	0.7936900000000000	0.3942600000000000	263.9154000000000
MVO [52]	0.7886027600000000	0.4084530700000000	263.8958499000000
GOA [57]	0.788897555578973	0.407619570115153	263.895881496069

TABLE 14: Comparison results for the car crashworthiness design problem.

Algorithm	RTLBO	DE [7]	GA [6]	FA [55]	CS [59]	GOA [57]	EOBL-GOA [58]
x1	0.50000	0.50000	0.50005	0.50000	0.50000	0.50000	0.50000
x2	1.11621	1.11670	1.28017	1.36000	1.11643	1.11670	1.11643
x3	0.50000	0.50000	0.50001	0.50000	0.50000	0.50000	0.50000
x4	1.30215	1.30208	1.03302	1.20200	1.30208	1.30208	1.30208
x5	0.50000	0.50000	0.50001	0.50000	0.50000	0.50000	0.50000
x6	1.50000	1.50000	0.50000	1.12000	1.50000	1.50000	1.50000
x7	0.50000	0.50000	0.50000	0.50000	0.50000	0.50000	0.50000
x8	0.34500	0.34500	0.34994	0.34500	0.34500	0.34500	0.34500
x9	0.332814	0.192000	0.192000	0.192000	0.192000	0.192000	0.192000
x10	-19.58840	-19.54935	10.31190	8.87307	-19.54935	-19.54935	-19.54935
x11	0.019066	-0.004310	0.001670	-18.998080	-0.004310	-0.004310	-0.004310
Optimal weight	22.84240	22.84298	22.85653	22.84298	22.84294	22.84474	22.84294

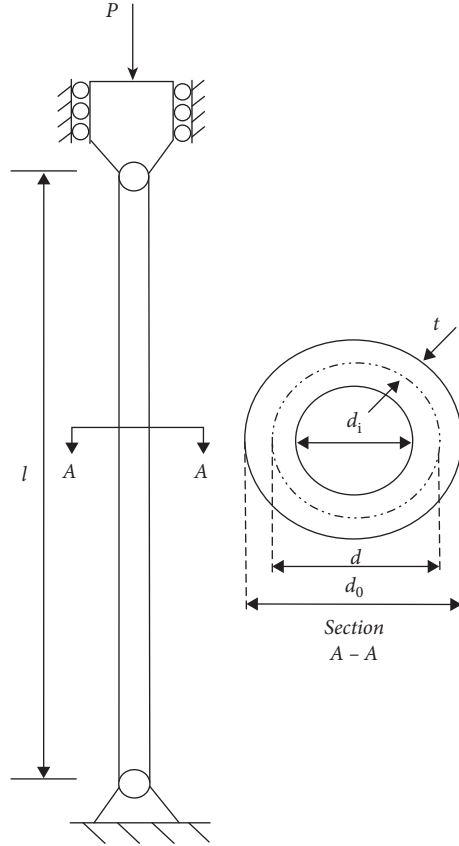


FIGURE 13: Tubular column design problem [59].

$$\left\{ \begin{array}{l}
 g_1 = \frac{P}{\pi d t \sigma_y} - 1 \leq 0, \\
 g_2 = \frac{8 P L^2}{\pi^3 E d t (d^2 + t^2)} - 1 \leq 0, \\
 g_3 = \frac{2.0}{d} - 1 \leq 0, \\
 g_4 = \frac{d}{14} - 1 \leq 0, \\
 g_5 = \frac{0.2}{t} - 1 \leq 0, \\
 g_6 = \frac{t}{0.8} - 1 \leq 0.
 \end{array} \right. \quad (20)$$

From the comparison results in Table 15, we can see that RLTLBO can obtain superior optimal cost compared to mGWO, DSCA, HOA, AO, HHO, and CS.

5.8. Frequency-Modulated Sound Waves Design Problem. This problem aims to optimize the frequency-modulated (FM) synthesizer parameter in six dimensions [60]. The following equation is given for optimization $X = \{a_1, \omega_1, a_2, \omega_2, a_3, \omega_3\}$ as a sound wave, where a_i ($i = 1, 2, 3$) is the amplitude and ω_i ($i = 1, 2, 3$) is the angular frequency. This problem has the lowest value $f(\vec{X}_{sol}) = 0$. The objective function is calculated based on the square errors between the target wave and the estimated wave. This problem is modeled as follows.

Minimize

$$f(\vec{X}) = \sum_{t=0}^{100} (y(t) - y_0(t))^2, \quad (21)$$

where

TABLE 15: Comparison results for the tubular column design problem.

Algorithm	Optimum variables		Optimum cost
	d	t	
RLTLBO	5.45120	0.29196	26.53130
mGWO	5.45080	0.29201	26.53270
DSCA	5.50250	0.29214	26.79030
HOA	5.26260	0.35487	28.86470
AO	5.46300	0.29656	26.83540
HHO	5.44380	0.29313	26.55820
CS [59]	5.45139	0.29196	26.53217

TABLE 16: Comparison results for the frequency-modulated sound waves design problem.

Algorithm	Optimum variables						Optimum cost
	$a1$	$\omega1$	$a2$	$\omega2$	$a3$	$\omega3$	
RLTLBO	-0.97498	-5.0327	-1.5640	-4.7840	-2.0060	4.9055	0.21738
GWO [13]	-0.66540	-0.1684	1.5173	-0.1287	-4.1335	-4.8997	8.47250
MFO [61]	0.61410	0.0432	-4.3251	4.7923	0.8339	0.1278	11.89690
PSO [11]	-0.58860	5.0145	-3.2779	-4.9324	-0.8562	-0.1476	13.18070
TSA [62]	0.34150	4.7881	1.4309	0.1158	0.0975	0.5480	25.10520
FFA [63]	-0.56270	0.0525	-3.4797	4.8930	1.1491	-4.8345	17.42910

$$\begin{cases} y(t) = a_1 \cdot \sin(\omega_1 \cdot t \cdot \theta + a_2 \cdot \sin(\omega_2 \cdot t \cdot \theta + a_3 \cdot \sin(\omega_3 \cdot t \cdot \theta))), \\ y_0(t) = (1.0) \cdot \sin((5.0) \cdot t \cdot \theta - (1.5) \cdot \sin((4.8) \cdot t \cdot \theta + (2.0) \cdot \sin((4.9) \cdot t \cdot \theta))), \\ \theta = \frac{2\pi}{100}. \end{cases} \quad (22)$$

The RLTLBO is compared with GWO, MFO [61], PSO, TSA [62], and FFA [63] algorithms, and the comparison results are listed in Table 16. It is obvious that the proposed method found a much better solution than the comparative algorithms.

In general, the excellent performance in solving industrial engineering design problems suggests that RLTLBO can be widely used in real-world optimization problems.

6. Conclusion

This study presents an improved teaching-learning-based optimization algorithm (RLTLBO) by incorporating reinforcement learning (RL) and random opposition-based learning (ROBL) strategies. Because of the defect of the insufficient learning processes, a new learning model is proposed in the learner phase. The two different modes uniting the inherent learning mode are switched through the Q-learning mechanism in RL. This mechanism helps the individuals learn thoroughly, resulting in accelerating the convergence speed of the RLTLBO. To improve the ability of local optima avoidance, the ROBL strategy is appended after the teacher and learner phases. The proposed RLTLBO algorithm is tested using 23 standard and eight CEC2017 benchmark functions to analyze its search performance. Experimental results illustrate competitive

results compared to other state-of-the-art meta-heuristic algorithms. To further verify the superiority of RLTLBO, eight industrial engineering design problems are solved. The results are also very competitive with other comparative algorithms.

The code for RLTLBO is provided at <https://github.com/WangShuang92/RLTLBO> and can be used for more practical problems. However, this algorithm still suffers with premature convergence on several benchmark functions, which can be studied in the future. Moreover, RLTLBO can only solve single objective problems. For future research, binary and multiobjective versions of RLTLBO can be considered. More applications of this algorithm in different fields are valuable works, including text clustering, scheduling problems, appliances management, parameter estimation, feature selection, test classification, image segmentation problems, network applications, sentiment analysis, etc.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

On behalf of all authors, the corresponding author states that there are no conflicts of interest.

Acknowledgments

This research was funded by National fund cultivation project of Sanming University (PYS2107 and PYT2105), the Sanming University Introduces High-level Talents to Start Scientific Research Funding Support Project (21YG01S, 20YG01, and 20YG14), Fujian Natural Science Foundation Project (2021J011128), Bidding project for higher education research of Sanming University (SHE2101), the Guiding Science and Technology Projects in Sanming City (2020-S-39, 2020-G-61, and 2021-S-8), the Educational Research Projects of Young and Middle-aged Teachers in Fujian Province (JAT200638 and JAT200618), and the Scientific Research and Development Fund of Sanming University (B202029 and B202009), Open Research Fund of Key Laboratory of Agricultural Internet of Things in Fujian Province (ZD2101), Ministry of Education Cooperative Education Project (202002064014), School level education and teaching reform project of Sanming University (J2010306 and J2010305), Higher education research project of Sanming University (SHE2102 and SHE2013), and 2021 project of the 14th Five-year Plan of Education science in Fujian Province (FJJKBK21-138).

References

- [1] L. Abualigah and A. Diabat, "Advances in sine cosine algorithm: a comprehensive survey," *Artificial Intelligence Review*, vol. 54, no. 4, pp. 2567–2608, 2021.
- [2] L. Abualigah and A. Diabat, "A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications," *Neural Computing & Applications*, vol. 32, no. 19, Article ID 15533, 2020.
- [3] W. Deng, X. Zhang, Y. Zhou et al., "An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems," *Information Sciences*, vol. 585, pp. 441–453, 2022.
- [4] Y. Kumar and P. K. Singh, "Improved cat swarm optimization algorithm for solving global optimization problems and its application to clustering," *Applied Intelligence*, vol. 48, no. 9, pp. 2681–2697, 2018.
- [5] E. Q. Wu, M. Zhou, D. Hu et al., "Self-paced dynamic infinite mixture model for fatigue evaluation of pilots' brains," in *IEEE Transactions on Cybernetics*, pp. 1–6, IEEE, 2020.
- [6] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.
- [7] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [9] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.
- [10] F. Asef, V. Majidnezhad, M.-R. Feizi-Derakhshi, and S. Parsa, "Heat transfer relation-based optimization algorithm (HTOA)," *Soft Computing*, vol. 25, no. 13, pp. 8129–8158, 2021.
- [11] J. Kennedy and R. Eberhart, "Particle swarm optimization," vol. 93, pp. 1942–1948, in *Proceedings of the 1995 IEEE International Conference on Neural Networks, IEEE ICNN*, vol. 93, IEEE, Perth, Australia, December 1995.
- [12] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
- [13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [14] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [15] L. Abualigah, D. Yousefi, M. A. Elaziz, A. A. Ewees, M. A. A. Alqaness, and A. H. Gandomi, "Aquila optimizer: a novel meta-heuristic optimization algorithm," *Computers & Industrial Engineering*, vol. 157, Article ID 107250, 2021.
- [16] H. Jia, X. Peng, and C. Lang, "Remora optimization algorithm," *Expert Systems with Applications*, vol. 185, p. 115665, 2021.
- [17] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, pp. 303–315, 2011.
- [18] A. Aouf, L. Boussaid, and A. Sakly, "TLBO-based adaptive neurofuzzy controller for mobile robot navigation in a strange environment," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 3145436, 8 pages, 2018.
- [19] M. Singh, B. K. Panigrahi, and A. R. Abhyankar, "Optimal coordination of directional over-current relays using teaching learning-based optimization (TLBO) algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 50, pp. 33–41, 2013.
- [20] D. L. Gonzalez-Alvarez, M. A. Vega-Rodriguez, J. A. Gomez-Pulido, and J. M. Sanchez-Perez, "Multiobjective teaching-learning-based optimization (MO-TLBO) for motif finding," in *Proceedings of the IEEE International Conference on Intelligence and Informatics CINTI'12*, pp. 1–6, IEEE, Budapest Hungary, November 2012.
- [21] F. Zou, D. Chen, and Q. Xu, "A survey of teaching-learning-based optimization," *Neurocomputing*, vol. 335, pp. 366–383, 2018.
- [22] Y. Kumar and P. K. Singh, "A chaotic teaching learning based optimization algorithm for clustering problems," *Applied Intelligence*, vol. 49, pp. 1036–1062, 2019.
- [23] A. Taheri, K. Rahimzadeh, and R. V. Rao, "An efficient balanced teaching-learning-based optimization algorithm with individual restarting strategy for solving global optimization problems," *Information Sciences*, vol. 576, pp. 68–104, 2021.
- [24] Y. Ma, X. Zhang, J. Song, and L. Chen, "A modified teaching-learning-based optimization algorithm for solving optimization problem," *Knowledge-Based Systems*, vol. 212, Article ID 106599, 2020.
- [25] Y. Xu, Z. Yang, X. Li, H. Kang, and X. Yang, "Dynamic opposite learning enhanced teaching-learning-based optimization," *Knowledge-Based Systems*, vol. 188, Article ID 104966, 2020.
- [26] H. Dong, P. Wang, and B. Song, "Kriging-assisted teaching-learning-based optimization (KTLBO) to solve computationally expensive constrained problems," *Information Sciences*, vol. 556, pp. 404–435, 2020.
- [27] Z. Ren, R. Jiang, F. Yang, and J. Qiu, "A multi-objective elitist feedback teaching-learning-based optimization algorithm

- and its application,” *Expert Systems with Applications*, vol. 188, Article ID 115972, 2021.
- [28] Y. Zhang, Z. Jin, and Y. Chen, “Hybrid teaching-learning-based optimization and neural network algorithm for engineering design optimization problems,” *Knowledge-Based Systems*, vol. 187, Article ID 104836, 2020.
- [29] A. V. Lakshmi and P. Mohanaiah, “WOA-TLBO: whale optimization algorithm with teaching-learning-based optimization for global optimization and facial emotion recognition,” *Applied Soft Computing*, vol. 110, Article ID 107623, 2021.
- [30] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, “Harris hawks optimization: algorithm and applications,” *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [31] F. Miarnaemi, G. Azizyan, and M. Rashki, “Horse herd optimization algorithm: a nature-inspired algorithm for high-dimensional optimization problems,” *Knowledge-Based Systems*, vol. 213, Article ID 106711, 2020.
- [32] S. Gupta and K. Deep, “A memory-based grey wolf optimizer for global optimization tasks,” *Applied Soft Computing*, vol. 93, Article ID 106367, 2020.
- [33] S. Wang, K. Sun, W. Zhang, and H. Jia, “Multilevel thresholding using a modified ant lion optimizer with opposition-based learning for color image segmentation,” *Mathematical Biosciences and Engineering*, vol. 18, pp. 3092–3143, 2021.
- [34] Y. Li, Y. Zhao, and J. Liu, “Dynamic sine cosine algorithm for large-scale global optimization problems,” *Expert Systems with Applications*, vol. 177, Article ID 114950, 2021.
- [35] E. Talbi, “Machine learning into metaheuristics: a survey and taxonomy of data-driven metaheuristics,” *ACM Computing Surveys*, vol. 54, pp. 1–32, 2020.
- [36] M. Drugan, “Reinforcement learning versus evolutionary computation: a survey on hybrid algorithms,” *Swarm and Evolutionary Computation*, vol. 44, pp. 228–246, 2019.
- [37] G. Lingam, R. R. Rout, and D. V. L. N. Somayajulu, “Adaptive deep Q-learning model for detecting social bots and influential users in online social networks,” *Applied Intelligence*, vol. 49, pp. 3947–3964, 2019.
- [38] F. Liu and G. Zeng, “Study of genetic algorithm with reinforcement learning to solve the TSP,” *Expert Systems with Applications*, vol. 36, pp. 6995–7001, 2009.
- [39] H. Samma, J. Mohamad-Saleh, S. A. Suandi, and B. Lahasan, “Q-learning-based simulated annealing algorithm for constrained engineering design problems,” *Neural Computing & Applications*, vol. 32, pp. 5147–5161, 2020.
- [40] Q. Chen, M. Huang, Q. Xu, H. Wang, and J. Wang, “Reinforcement learning-based genetic algorithm in optimizing multidimensional data discretization scheme,” *Mathematical Problems in Engineering*, vol. 20, pp. 1–13, 2020.
- [41] Y. Xu and D. Pi, “A reinforcement learning-based communication topology in particle swarm optimization,” *Neural Computing & Applications*, vol. 32, pp. 1–26, 2020.
- [42] E. Emary, H. M. Zawbaa, and C. Grosan, “Experienced gray wolf optimization through reinforcement learning and neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 681–694, 2019.
- [43] M. Ghafoorian, N. Taghizadeh, and H. Beigy, “Automatic Abstraction in Reinforcement Learning Using Ant System Algorithm,” in *Proceedings of the AAAI Spring Symposium Series*, pp. 9–14, Stanford, CA, USA, March 2017.
- [44] A. Seyyedabbasi, R. Aliyev, F. Kiani, M. U. Gulle, and M. A. Shah, “Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems,” *Knowledge-Based Systems*, vol. 222, Article ID 107044, 2013.
- [45] H. Tizhoosh, “Opposition-based learning: a new scheme for machine intelligence,” in *Proceedings of the International Conference on Computational Intelligence for Modelling*, pp. 695–701, Vienna, Austria, November 2005.
- [46] W. Long, J. Jiao, X. Liang, S. Cai, and M. Xu, “A random opposition-based learning grey wolf poptimizer,” *IEEE Access*, vol. 7, Article ID 113810, 2019.
- [47] X. Yao, Y. Liu, and G. Lin, “Evolutionary programming made faster,” *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 82–102, 1999.
- [48] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm Evolut Comput*, vol. 1, pp. 3–18, 2011.
- [49] N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang, and B. Y. Qu, “Problem Definitions and Evaluation Criteria for the CEC2017,” in *Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*, IEEE Congress on Evolutionary Computation, 2016.
- [50] S. M. Li, H. L. Chen, M. J. Wang, A. A. Heidari, and S. Mirjalili, “Slime mould algorithm: a new method for stochastic optimization,” *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.
- [51] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, “Marine predators algorithm: a nature-inspired metaheuristic,” *Expert Systems with Applications*, vol. 152, Article ID 113377, 2020.
- [52] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-verse optimizer: a nature-inspired algorithm for global optimization,” *Neural Computing & Applications*, vol. 27, pp. 495–513, 2015.
- [53] Z. W. Geem, J. H. Kim, and G. V. Loganathan, “A new heuristic optimization algorithm: harmony search,” *SIMULATION*, vol. 76, pp. 60–68, 2001.
- [54] I. Rechenberg, “*Evolutionsstrategien*,” Springer Berlin Heidelberg, Berlin, Germany, pp. 83–114, 1978.
- [55] S. Mirjalili, “SCA: a sine cosine algorithm for solving optimization problems,” *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.
- [56] G. A. Baykaso and F. B. Ozsoydan, “Adaptive firefly algorithm with chaos for mechanical design optimization problems,” *Applied Soft Computing*, vol. 36, pp. 152–164, 2015.
- [57] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper optimization algorithm: theory and application,” *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.
- [58] B. S. Yildiz, N. Pholdee, S. Bureerat, A. R. Yildiz, and S. M. Sait, “Enhanced Grasshopper Optimization Algorithm Using Elite Opposition-Based Learning for Solving Real-World Engineering Problems,” *Engineering with Computers*, 2021.
- [59] A. H. Gandomi, X. S. Yang, and A. H. Alavi, “Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,” *Engineering with Computers*, vol. 29, pp. 17–35, 2013.
- [60] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, “Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems,” *International Journal of Intelligent Systems*, vol. 36, pp. 1–72, 2021.
- [61] S. Mirjalili, “Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm,” *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

- [62] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 90, pp. 1–29, 2020.
- [63] H. Shayanfar and F. S. Gharehchopogh, "Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems," *Applied Soft Computing*, vol. 71, pp. 728–746, 2018.