Strengthened Teaching–Learning-Based Optimization Algorithm for Numerical Optimization Tasks

Xuefen Chen
University of Shanghai for Science and Technology

Chunming Ye (✉️ 15902171827@163.com)
University of Shanghai for Science and Technology

Yang Zhang
University of Shanghai for Science and Technology

Lingwei Zhao
University of Shanghai for Science and Technology

Jing Guo
University of Shanghai for Science and Technology

Kun Ma
University of Shanghai for Science and Technology

Research Article

Keywords: Metaheuristic, optimization algorithm, Teaching–learning-based optimization algorithm, teaching factor, elite system, Cauchy mutation

Posted Date: November 15th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-696824/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
Strengthened Teaching–Learning-Based Optimization Algorithm for numerical optimization tasks

Xuefen Chen, Chunming Ye*, Yang Zhang, Lingwei Zhao, Jing Guo, Kun Ma

Abstract
The teaching–learning-based optimization algorithm (TLBO) is an efficient optimizer. However, it has several shortcomings such as premature convergence and stagnation at local optima. In this paper, the strengthened teaching–learning-based optimization algorithm (STLBO) is proposed to enhance the basic TLBO’s exploration and exploitation properties by introducing three strengthening mechanisms: the linear increasing teaching factor, the elite system composed of new teacher and class leader, and the Cauchy mutation. Subsequently, seven variants of STLBO are designed based on the combined deployment of the three improved mechanisms. Performance of the novel STLBOs is evaluated by implementing them on thirteen numerical optimization tasks. The results show that STLBO7 is at the top of the list, significantly better than the original TLBO. Moreover, the remaining six variants of STLBO also outperform TLBO. Finally, a set of comparisons are implemented between STLBO7 and other advanced optimization techniques. The numerical results and convergence curves prove that STLBO7 clearly outperforms other competitors, has stronger local optimal avoidance, faster convergence speed and higher solution accuracy. All the above manifests that STLBOs has improved the search performance of TLBO. Data Availability Statements: All data generated or analyzed during this study are included in this published article.

Keywords Metaheuristic · optimization algorithm · Teaching–learning-based optimization algorithm · teaching factor · elite system · Cauchy mutation

1 Introduction

In the past decades, meta-heuristic algorithms are widely applied in various research fields and practical scenarios, which are favored by researchers because of their simplicity, efficiency, low computational cost, and extraordinary performance. Different from exact algorithms, meta-heuristic algorithms are a type of optimization technology that seeks approximate optimal solutions of the problem under limited time and resource constraints. Nowadays, with the development of emerging technologies such as the internet and artificial intelligence, large-scale optimization tasks with features such as non-differentiable, non-differentiable, or discontinuities are increasing rapidly. This allows meta-heuristic algorithms to play an increasingly important role. In meta-heuristics, exploration and exploitation abilities are two crucial characteristics, which together affect the search performance of the algorithm.

1.1 Related work

Dividing meta-heuristic algorithms into single-point based search and multi-point based search is the simplest and most common classification method. In the literature [1], these algorithms can be divided into three categories based on their source of inspiration, namely swarm intelligence algorithms, non-swarm intelligence algorithms, and physical chemical algorithms. In the paper [2], a new classification method based on whether specific parameters need to be set in addition to general parameters is proposed, which classifies meta-heuristic algorithms into two groups: special parameter meta-heuristic algorithms and free-special parameter meta-heuristic. In these sub-categories, population-based intelligence optimization algorithms (i.e., swarm intelligence algorithms) occupy a critical and irreplaceable position. As we all know, nature has become the main source of inspiration for the swarm intelligence optimizer in recent years. Generally, the optimization process starts with randomly generating a set of solutions in the search domain, and then combines, reorganizes, moves, and evolved through a preset calculation mechanism, and finally gradually approaches the global optimal solution [3].

Population-based intelligence algorithms are widely known. Some recognized swarm intelligence optimization algorithms inspired by nature are mainly include: genetic al-
formed by applying the big bang theory in physics combining EOA [14], salp swarm algorithm (SSA) [15], elite genetic chasing styles when killing prey. SSO simulates the dynamic inspired Harris hawk's cooperative hunting behaviors and the concepts of black hole, white hole and wormhole. HHO is rotate around the artificial light source. HS is established by PSO is designed based on the predatory behavior of birds. SSA [11]. These algorithms are inspired by different natural phenomena. Among them, GA is presented based on Darwin's theory of biological evolution. PSO is designed based on the predatory behavior of birds. SSA [11]. These algorithms are inspired by different natural phenomena. Among them, GA is presented based on Darwin's theory of biological evolution. MVO is constructed by simulating the behavior that the moths rotate around the artificial light source. HS is established by considering the process of musicians playing chords. MVO is formed by applying the big bang theory in physics combing the concepts of black hole, white hole and wormhole. HHO is inspired Harris hawk's cooperative hunting behaviors and chasing styles when killing prey. SSO simulates the dynamic foraging strategy of southern flying squirrels and their efficient locomotion way. EHO is proposed by modeling the unique live behaviors of the elephants belonging to different clans. Other population-based optimizers include: gravitational search algorithm (GSA) [12], monarch butterfly optimization algorithm (MBO) [13], earthworm optimization algorithm (EOA) [14], salp swarm algorithm (SSA) [15], elite genetic algorithm with tabu search (TS-EGA) [16], krill herd algorithm (KH) [17], and so on. These naturally inspired population-based optimizers have been proved to be efficient. However, according to the No Free Lunch (NFL) theorem, no optimizer is suitable for solving all the problems in the world. Therefore, Rao et al. developed a new population-based optimization method called teaching–learning-based optimization algorithm (TLBO) by modeling human teaching-learning behaviors [18].

TLBO is a novel population-based optimization method, inspired by teaching behavior and teaching mode in human society. The design concept of TLBO is that the teacher in the class is keen to improve the average score of the whole class to his/her own level. Moreover, the students in the class try to increase their knowledge by learning from the teacher and communicating with each other. Ideally, each member of the class should perform his or her duties to make the whole group evolve in a better direction.

TLBO is first proposed to solve constrained mechanical design optimization, continuous non-linear function, and engineering optimization problems. Subsequently, it is used in more and more academic research and practical application fields. For instance, in the literature [19], TLBO is used to optimize energy consumption and surface defects in wire cut electric discharge machining. The results reveal that the surface defects around the slit are the least under the optimized working conditions of TLBO. In the work [20], a functional link artificial neural network based TLBO is proposed for real-time identification of fuzzy PID-control magnetic levitation system. In the research [21], a modified binary TLBO (MBTLBO) is designed for features subset selection. The experiments results show the MBTLBO can increase the classification accuracy of risk features with rheumatism disease.

Although TLBO performs well, it still has the defects of immature convergence, slow convergence speed, low efficiency, and stagnation at local optima for some complex optimization tasks. Therefore, many scholars are committed to introducing various improvement mechanisms to improve the search performance of conventional TLBO. For example, Rao et al. proposed an improved TLBO by deploying the concept of number of teachers, adaptive teaching factor, tutorial training and self-motivated learning [22]. Niu et al. presented an improved TLBO with elite strategy for parameters identification [23]. Sultana et al. designed a quasi-oppositional TLBO named QOTLBO by introducing the concepts of opposition-based learning (OBL) and quasi OBL [24]. Recently, Jiang et al. proposed an improved TLBO called NFDR-TLBO by adding the neighborhood topology mechanism for multi-level thresholding image segmentation [25].

Based on the above literature review, it has theoretical and practical significance for the study of TLBO. In this paper, strengthened teaching-learning-based optimization algorithms (SGWOs) is proposed by setting up three strengthening mechanisms in Section 3: a linear increasing teaching factor, an elite system, and a Cauchy mutation strategy in learner phase. Subsequently, a total of seven STLBOs are constructed based on permutations and combinations of three strengthening mechanisms.

The source of inspiration for the improvement mechanisms in this work is as follows. Firstly, Zhang et al. proposed a modified Harris hawk optimization algorithm named MHHO by changing the update method of jumping energy [2]. Zhan et al. designed an adaptive particle swarm optimization algorithm called APSO by automatic control of inertia weight, acceleration coefficient and other algorithm parameters [26]. It can be seen that the key parameters in the optimization algorithm have a greater impact on the performance of the algorithm. Therefore, we reset the update mechanism of teaching factors to better meet the requirements of reality and algorithms. Secondly, we have joined an elite system of the new teacher and class leader in the learning phase to accelerate convergence and learn more knowledge for guide the correct evolution of the population. Thirdly, the authors proposed a fast evolutionary programming (FEP) by introducing the
mechanism of Cauchy mutation operator [27]. The results reveal that EFP has excellent search performance. Therefore, we added the Cauchy mutation mechanism to the original TLBO to enhance population diversification for better jump out of the local optimum.

1.2 Contributions

The contribution of this paper is threefold. Firstly, a novel population-based optimization algorithm call STLBO and seven variants of STLBOs are proposed. The three strengthening mechanisms is introduced, namely the linear increasing teaching factor, the elite system, and the Cauchy mutation. Secondly, comparative experiments between TLBO and STLBOs are carried out. The numerical results show that seven STLBOs can enhance the performance of TLBO. Among them, STLBO7 with the three mechanisms is obviously superior to other rivals as the best competitor. Thirdly, comparative experiments organized in Section 4.3 between STLBO7 and other famous technologies are carried out. The comparison between the numerical results and the convergence curve further illustrates the superiority of STLBO7.

Organization: The rest of this paper is organized as follows. Section 2 describes the original TLBO method. Section 3 presents the latest STLBOs mechanisms. Section 4 lists the experimental results and related analysis. Finally, the summary and some associated revelations are showed in Sect. 5.

2 TLBO algorithm

Undoubtedly, human beings are regarded as the most intelligent creatures in nature because they can acquire rich knowledge and skills through continuous learning. In human society, school education is a critical way for everyone to improve themselves and grow up. In a class, educator refer to the teachers, while the educated refer to students. The teachers and students complete the teaching task together through communication. This process is mainly based on teachers imparting knowledge and skills, while students' goal is to accept as much knowledge and skills as possible. However, in the actual scene, due to the individual differences between students and the combined effect of other influencing factors, different students have different degrees of acceptance and mastery of knowledge. Generally, the subject examination results is regarded as an important indicator to evaluate students’ learning effect and teachers' teaching level within a time span. Therefore, in the whole teaching cycle, teachers strive to improve the average score of the class by designing courses, designing teaching plans, implementing teaching activities and reflection, while students also strive to improve their performance by studying hard and communicating with other individuals, so as to realize the dynamic optimization of the whole group.

Inspired by the human teaching–learning process, Rao et al. proposed a new population-based meta-heuristic algorithm named TLBO based on the teaching and learning behavior between teachers and students in a class. The design idea of TLBO mainly comes from the construction of a typical optimization system (namely teaching–learning activities). In this system, a class in a school is regarded as a population in swarm intelligence optimization algorithm, students and teachers are considered as search agents, and their test scores or knowledge level are treated as fitness.

For a clearer explanation, the distribution of the students’ grades in the two classes taught by $T_1$ and $T_2$ after the implementation of a teaching activity is shown in the Fig. 1, and the distribution of students’ scores before and after the implementation of teaching activities in a class is shown in Fig. 2. As we all know, the distribution of achievements satisfies the normal distribution expressed by Equation (1). As shown in Fig. 1, the curve-1 and curve-2 represent distribution of grades obtained by students taught by $T_1$ and $T_2$, respectively. Obviously, the teaching level of $T_2$ is better than that of $T_1$. Therefore, teachers play an important role in the evolution of class population.

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$ (1)

where $\sigma^2$ is the variance, $\mu$ is the mean.

Fig. 1. Distribution of grades obtained by students taught by two different teachers.

Fig. 2. Distribution of grades obtained by students taught by a teacher.
The TLBO contains two stages: “Teacher phase” and “Learner phase”. The specific implementation details are showed in the following.

2.1 Teacher phase

In the teacher stage, the best individual with the optimal fitness in the class is designated as the teacher. The purpose of selecting teacher is to impart knowledge and try to improve the average grade of the class to the teacher level. In this situation, each decision variable can be regarded as a subject, and the individual’s fitness can be regarded as his/her total score in all subjects. However, it is almost impossible for teacher to fully achieve this goal because of many factors such as students’ uneven learning and comprehension ability. Therefore, a concept called teaching factor \( T_F \), calculated by Equation (3) is introduced to conform to the facts.

Suppose that the current iteration number is \( k \), and let \( M_k \) is the average score of all individuals, \( T_k \) is the teacher at iteration \( k \). The difference between \( M_k \) and \( T_k \) can be expressed by Equation (2).

\[
\text{Difference} = r_k \times (T_k - T_F \times M_k)
\]
\[
T_F = \begin{cases} 
1, & r \leq 0.5 \\
2, & r > 0.5 
\end{cases}
\]

(1) \hspace{1cm} (2)

where \( r_k \) and \( r \) satisfy the uniform distribution of [0,1].

Subsequently, the position of the current solution can be updated based on the difference, as shown in Equation (4).

\[
X_{\text{new},i}^{k} = X_i^{k} + \text{Difference}
\]

(4)

where \( X_i^{k} (i = 1,2,...,N) \) is the position the \( i \)th search agent before teaching activities, and \( X_{\text{new},i}^{k} \) is the position the \( i \)th search agent after teaching activities in iteration \( k \). \( N \) indicates the preset population size.

2.2 Learner phase

Learners (students) increase knowledge in two different ways: one is through the teacher’s teaching, the other is through their interaction. In the learner stage, students improve their performance by communicating with each other. The communication methods are mainly conducted through group discussion, personal experience introduction, formal or informal conversation, etc. In communication, students can improve their knowledge by random interaction with others. If a student finds that another student has more knowledge than he/she, he/she can learn something new. The specific individual renewal formula is as follows.

For the individual \( X_i^{k} \) after the teaching stage, another student \( X_j^{k'} (i \neq j) \) was randomly selected. If \( f(X_i^{k'}) < f(X_i^{k}) \), the position update formula of \( X_i^{k'} \) can be expressed as:

\[
X_i^{k+1} = X_i^{k'} + r_i(X_i^{k'} - X_j^{k'})
\]

(5)

Otherwise,

\[
X_i^{k+1} = X_i^{k} + r_i(X_i^{k} - X_j^{k})
\]

(6)

Finally, the fitness was reevaluated. If the fitness of \( X_i^{k+1} \) is better than that of \( X_i^{k'} \), the update will be accepted; otherwise, keep \( X_i^{k'} \) unchanged.

2.3 Pseudocode of TLBO

In this subsection, the pseudocode of TLBO is presented in Algorithm 1.

Algorithm 1 Pseudocode of TLBO algorithm

1: Inputs: \( N \) and total number of iteration.
2: While (termination condition is not met)
3: Calculate the mean \( M \)
4: Calculate difference: Difference
5: for \( i = 1: N \) % Teacher phase
6: \( X_i^k = X_i^k + \text{Difference} \)
7: \( X_{\text{new},i}^k = X_i^k \)
8: for \( i = 1: N \) % Learner phase
9: Randomly select another learner \( X_j^k \)
10: \( X_{\text{new},i}^k \) is a learner
11: End if
12: End for
13: End while
14: Accept \( X_{\text{new},i}^k \)
15: End for
16: Return global best solution

3 MG-TLBO algorithm

In this section, strengthened teaching–learning-based optimization algorithm called STLBOs is proposed based several improvements to the original TLBO. In the STLBO, three reinforcement mechanisms are introduced to improve the searching performance of the conventional TLBO.
3.1 A linear increasing teaching factor

In the teacher phase, teaching factor $T_F$ is a unique variable that can decide the mean value to be changed. $T_F$ is assigned equal probability to 1 or 2. However, after full and detailed numerical experiments on $T_F$, we find that the value of 3 or 4 may be a more desirable choice. From a practical point of view, compared with 1 or 2, $T_F$ value of 3 or 4 means that teachers have a higher level of teaching, while students have a higher learning ability, and can even learn the knowledge that teachers have not taught. From the perspective of algorithm, when $T_F$ is 3 or 4, the search agent can approach the global optimal solution faster, so the algorithm has higher convergence speed. Further, in the actual scenario, with the development of teaching activities, the teaching level of teachers should be gradually improved rather than randomly assigned. Therefore, a linear increasing $T_F$ is designed to model its adaptive changes with iterations in this subsection, as shown in Equation (7).

$$T_F = 3 + k/T$$  \hspace{1cm} (7)

where $k$ is the current number of iterations and $T$ represents the maximum number of iterations.

3.2 Elite system in learner phase

In the learner phase of the conventional TLBO, class members increase their knowledge by communicating and learning with each other. However, this learning method has certain shortcomings. The most important point is that class members lack the guidance of outstanding individuals. This kind of mutual communication and learning is messy, without certain benchmarks and standards, so it may cause a lot of useless "learning". From the perspective of population evolution, the existence of this invalid learning strategy will make the population evolve in the wrong direction to a certain extent.

We constructed an elite system in the learning phase in order to reasonably solve the above algorithm defects. In this subsection, an elite system is introduced into TLBO to strengthen the role of elite individuals in each iteration. In the elite system, a new teacher ($T_{\text{new}}$) and a class leader ($X_{\text{leader}}$) are selected from the new class that has passed the teacher stage. In the implementation of the algorithm, $T_{\text{new}}$ and $X_{\text{leader}}$ respectively represent the elite individuals with the first and second fitness in the current population. They have the most abundant information about the global optimal solution and together constitute the elite system. $T_{\text{new}}$ and $X_{\text{leader}}$ ensure that students can learn as much useful knowledge as possible. Meanwhile, the setting of $X_{\text{leader}}$ ensures the diversity of the population to effectively avoids the local optimal solution.

The elite system introduced into the learner phase of TLBO can be expressed as:

- $T_{\text{new}} = \text{the best search agent in current iteration } k$.
- $X_{\text{leader}} = \text{the second best search agent in current iteration } k$.

In the implementation of the algorithm, the remaining search agents update their positions based on $T_{\text{new}}$ and $X_{\text{leader}}$ with the same probability. Let $r$ be a random number between [0, 1], if $r$ is greater than 0.5, the position update formula can be expressed as Equation (8); otherwise, the position update formula can be expressed as Equation (9).

$$X_{i}^{k+1} = X_{i}^{k} \pm r_{i}(T_{\text{new}} - X_{i}^{k})$$  \hspace{1cm} (8)

$$X_{i}^{k+1} = X_{i}^{k} \pm r_{i}(X_{\text{leader}} - X_{i}^{k})$$  \hspace{1cm} (9)

where, $X_{i}^{k}$ denotes the position vector of the ith individual after the teacher phase, and $X_{i}^{k+1}$ denotes the position vector of the ith individual after the learner phase. $r_{i}$ denotes a random number between [0, 1].

3.3 A Cauchy mutation mechanism

Although TLBO is a superior optimizer, it has depicts such as premature convergence and weak local optimal avoidance when dealing with some complex tasks. To solve this problem, it is extremely necessary to introduce a certain mutation mechanism.

In this subsection, a Cauchy mutation strategy (CM) is integrated into the learner phase of the TLBO. Generally, the mutation mechanism improves the search performance of the search agent by enhancing the diversity of the population. In this paper, we use the Cauchy distribution to model Cauchy mutation mechanism. The mathematical formula of CM calculation can be expressed as $CM \sim \mathcal{C}(\gamma, x_0)$, $\gamma = 0.5, x_0 = 1$.

The probability density function of the Cauchy distribution is shown in Equation (10).

$$f(x; x_0, \gamma) = \frac{1}{\pi \gamma} \left(\frac{x}{(x - x_0)^2 + \gamma^2}\right)$$  \hspace{1cm} (10)

where $x_0$ is the location parameter defining the peak position of the distribution; $\gamma$ is the scale parameter of the half width at half of the maximum value.

When $\gamma = 0.5$ and $x_0 = 1$, the probability density function image of Cauchy distribution is shown in Fig. 3.

In the implementation of the algorithm, a $CM_{\text{factor}}$ is designed to model CM mechanism.

$$CM_{\text{factor}} = R A N D \ast CM$$  \hspace{1cm} (11)

where RAND represents the D-dimensional vector with value between [0.01, 0.001], and D is the number of variables. $CM$ represents the D-dimensional distribution number vector.

Then, the new location update combined CM mechanism is expressed by Equation (12).
where $X_i^{k+1'}$ indicates the location of the search agent after the implementation of the CM calculation.

\[ X_i^{k+1'} = X_i^k \times CM_{factor} \]  

(12)

Fig. 3. Probability density function of Cauchy distribution.

3.4 Schematic diagram of STLBO

In this subsection, the pseudocode of STLBO deployed with the above three mechanisms is presented in Algorithm 2.

**Algorithm 2. Pseudocode of STLBO algorithm**

1. **Inputs:** N and total number of iteration.
2. While (termination condition is not met) do
3. Calculate the mean M
4. Calculate difference using Equation (7)
5. for $i = 1$ to N do % Teacher phase
6. $X_{new,i}^k = X_i^k + \text{Difference}$
7. If $f(X_{new,i}^k) < f(X_i^k)$ then
8. $X_i^k = X_{new,i}^k$
9. Else
10. $X_i^k = X_i^{k'}$
11. End if
12. End for
13. Select $T_{new}$ and $X_{leader}$ % Learner phase
14. for $i = 1$ to N do % Learner phase
15. $r = \text{rand}()$
16. If $r > 0.5$ then
17. $X_i^{k+1} = X_i^{k'} + r(T_{new} - X_i^{k'})$
18. $X_i^{k+1'} = X_i^{k+1} \times CM_{factor}$
19. Else if $r < 0.5$ then
20. $X_i^{k+1} = X_i^{k'} + r(X_{leader} - X_i^k)$
21. $X_i^{k+1'} = X_i^{k+1} \times CM_{factor}$
22. End
23. If $f(X_i^{k+1}) < f(X_i^{k'})$ then
24. Accept $X_i^{k+1}$
25. End if
26. End for
27. End while
28. Return global best solution

4 Experiment and analysis

In this section, the two experiments are carried out to measure the performance of the STLBOs. First, a comparative experiment is performed on seven STLBO variants and the conventional TLBO. The description of the algorithm is shown in Table 1. Then, the SGWO7 with the best search performance is selected in the next experiments. All experimental results and analysis are recorded and compared with several famous optimization technology such as HS, PSO, MFO, GA and traditional TLBO.

In this paper, the total number of iterations in comparison methods are set to 1000 except for the HS. The population size is set to 40 for PSO, MFO, and GA, and the population size is set to 20 for TLBO and STLBOs. This is because TLBO and STLBOs have $2 \times N$ (Population Size) function evaluations in one iteration, which is twice that of PSO, MFO, and GA. For HS, the best population size is 5-7. Therefore, we set the population size of HS to 6. Meanwhile, to ensure the same total number of function evaluations, the total iterations are set to 40,000, same as other algorithms. Detailed parameter settings are listed in Table 2.

As shown in Table 2, in addition to the two general parameters (i.e., population size and total iterations), some special parameters need to be set for HS, PSO, and GA. For HS, the HMCR, PAR, and BW indicates the harmony memory considering the rate, pitch-adjustment rate, and bandwidth, respectively. For PSO, $\nu_{\text{max}}$ refer to the maximum speed of the particle moving. $w_1$ and $w_2$ are the initial and final weights, $c_1$ and $c_2$ represent the cognitive and social parameters, respectively. For GA, the chromosome replication, crossover and mutation strategies adopt roulette replication, uniform crossover, and random selection mutation respectively. $p_{\text{crossover}}$ indicates chromosome crossover mutation probability, and $p_{\text{mutation}}$ indicates chromosome mutation probability.

The experiments were implemented on a Windows 10 operating system with Inter(R) CPU i7-6700HQ @2.60 GHz and 4.00-GB RAM. All algorithms were encoded using Python 3.7.

(13 to be inserted here.)

(14 to be inserted here.)

4.1 Test Functions

In this study, thirteen well-known test functions, including seven unimodal (f1-f7) and six multimodal (f8-f13) functions [28], are selected for our experiments. These functions can be used to effectively measure the search performance of meta-heuristic algorithms. Among them, the unimodal benchmark function is an indicator of the convergence speed and solution accuracy of the optimization algorithm because it has only a strict local peak in the search domain. In other words, the numerical result on the unimodal function is equivalent to the algorithm exploitation characteristics. Unlike the unimodal function, the multimodal function refers to a test function that has several local optimal solutions in the search space. Obvi-
ously, these multimodal functions can be used to measure the algorithm exploration properties. For multi-peak tasks, how to effectively avoid local optima and accurately locate the global optima is the first priority of the optimization algorithm. These benchmark functions quantify the exploitation and exploration nature of algorithms, and provide mathematical methods for comparison and analysis. The mathematical description of these functions is listed in Tables 3 and 4.

4.2 Comparative test of seven STLBOs with TLBO

In this subsection, the comparison experiments between TLBO and seven STLBOs variants on a series of benchmark functions are implemented. The comparison algorithms and their corresponding descriptions are shown in Table 1. We choose two dimensions (i.e., 10D, and 30D) of benchmark functions for experiments. The numerical results are recorded in Tables 5 and 6. To ensure the robustness of the results, the numerical experiments are independently run 30 times, and the mean (Mean) and the standard deviation (STD) of the 30 runs are taken as evaluation indicators. In the tables, "Rank" represents the ranking of the comparison methods on per benchmark function. The Boldface indicates the best results obtained by this optimizer.

Seen from Tables 5 and 6, STLBO2, STLBO4, STLBO6, and STLBO7 obtain the theoretical optimal solution (0) for functions F1 and F3. Additionally, the variance of 0 indicates that the result is quite stable. The optimal results is achieved by STLBO4 for functions F2 and F4. For function F5, STLBO3 is the best optimization technique. For function F6, when the dimension is 10D, STLBO3 has achieved the most excellent result, followed by STLBO7; when the dimension is 30D, STLBO5 has achieved the most excellent result. When the dimension is 10D, STLBO4 is the brightest competitor for functions F7 and F8. However, when the dimension is 30D, the best results is got by STLBO7 and STLBO6 for functions F7 and F8. For functions F9 and F11, the all optimizer achieve the global best solution, which shows that TLBOs performs outstandingly for the two tasks. For function F10, the experimental results of STLBO2, STLBO4, STLBO6, and STLBO7 are tied for first place. For function F12, when the dimensions are 10D and 30D, STLBO2 and STLBO4 are the best methods respectively. For function F13, STLBO3 beats the other rivals when the dimension is 10D; STLBO7 win the first place when the dimension is 30D.

In the tables, “Average Rank” and “Overall Rank” provide a more intuitive indicator for algorithm comparison. From Table 5, STLBO7 presents outstanding performance, followed by STLBO4 and STLBO6. From Table 6, STLBO7 is still achieved an overwhelming victory over opponents, followed by STLBO6 and STLBO4. Finally, the total ranking table (Table 7) of the comparison methods is obtained by combining the results of Tables 5 and 6. It can be seen from Table 7 that STLBO7 came out on top, compared with other algorithms. Moreover, it can be seen that the original TLBO ranks last. Therefore, all the enhancement strategies proposed in this paper are effective, and the search performance of the STLBO that introduces the improved mechanism in Section 3 will improve. The seven STLBO variants are effective, and different problems are suitable for different STLBOs. Overall, STLBO7 is the most successful competitor and it was selected for subsequent experiments.

4.3 Comparative test of STLBO7 with other optimizer

In this subsection, some other prestigious optimizers such as HS, PSO, MFO, and GA are selected for comparison experiments. The comparison algorithms and their corresponding parameter settings are list in Table 2. We choose three dimensions (i.e., 40D, 70D, and 100D) of benchmark functions for experiments. The experiment results are recorded in Tables 8-10.

Based on Tables 8-10, it is seen that newly proposed STLBO7 obtains the most competitive solutions than other comparison optimizers by analyzing the mean and STD for nine functions (i.e., functions F1-F7, F10, and F13) in selected dimension. Among them, STLBO7 reach the theoretical optimal value for functions F1 and F3. For functions F8, GA can get the best average fitness compared with other algorithms, followed by HS and STLBO7. Although STLBO7 is not satisfactory in solving F8, its numerical results are still significantly better than the original TLBO., STLBO7 and TLBO, with zero Mean and zero Std, gains a slight advantage For functions F9 and F11. For function F12, HS has obvious advantages over other methods when the dimension is 40D; GA is the best optimization technology when the dimensions are 70D and 100D, followed by STLBO7.

It can be perceived from the “The total ranking table” (Table 11) that STLBO7 with “Total Average Rank” (1.2821) is the most outstanding one among all comparison methods. Therefore, the newly designed STLBO7 is an efficient optimization technology.

(Table 3 to be inserted here.)
(Table 5 to be inserted here.)
(Table 6 to be inserted here.)
(Table 7 to be inserted here.)
(Table 8 to be inserted here.)
(Table 9 to be inserted here.)
(Table 10 to be inserted here.)
(Table 11 to be inserted here.)
The convergence curves of the comparison optimizers on the 70D functions is shown in Figs. 4 and 5 to further illustrate the effectiveness of the proposed STLBO7. From Fig. 4, it can be perceived that compared with the traditional TLBO and other optimization techniques, STLBO7 has higher solution accuracy and the fastest convergence speed on all unimodal problems. From Fig. 5, STLBO7 is still far better than other optimizers in solving accuracy and convergence speed on multimodal benchmark functions other than function F8. For function F8, GA gets the highest solution accuracy and fastest convergence speed. Although TLBO and STLBO7 cannot find the global optimal solution of F8 within 500 iterations, the latter still has better searching performance than the former. These curves further illustrate the effectiveness of our proposed STLBO7.

(Fig. 4 to be inserted here.)
(Fig. 5 to be inserted here.)

Table 1 The description of TLBO and seven STLBOs

| Algorithm   | Description                          |
|-------------|--------------------------------------|
| TLBO        | Standard TLBO                        |
| STLBO1      | STLBO using improved strategy in Section 3.1 |
| STLBO2      | STLBO using improved strategy in Section 3.2 |
| STLBO3      | STLBO using improved strategy in Section 3.3 |
| STLBO4      | STLBO using improved strategies in Sections 3.1 and 3.2 |
| STLBO5      | STLBO using improved strategies in Sections 3.1 and 3.3 |
| STLBO6      | STLBO using improved strategies in Sections 3.2 and 3.3 |
| STLBO7      | STLBO using improved strategies in Sections 3.1 and 3.2 and 3.3 |

Table 2 The algorithm parameters

| Optimizer | Population size | Total Iterations | Unique parameters |
|-----------|-----------------|------------------|-------------------|
| HS        | 6               | 40000            | HMCR = 0.9, PAR = 0.3, BW = 0.1 |
| PSO       | 40              | 1000             | v_{max} = 20, w_1 = 0.9, w_2 = 0.4, c_1 = 1.2, c_2 = 1.2 |
| GA        | 40              | 1000             | p_{crossover} = 0.7, p_{mutation} = 0.3 |
| MFO       | 40              | 1000             | / |
| TLBO      | 20              | 1000             | / |
| STLBOs    | 20              | 1000             | / |

Table 3 The unimodal benchmark functions

| Function | Dimensions | Range          | Global optima |
|----------|------------|----------------|---------------|
| f_1(x) = \sum_{i=1}^{n} x_i^2               | 30,60,100     | [-100,100]     | 0             |
| f_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|     | 30,60,100     | [-10,10]      | 0             |
| f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i-1} x_j)^2        | 30,60,100     | [-100,100]     | 0             |
| f_4(x) = \max_{i \leq n} |x_i|       | 30,60,100     | [-100,100]     | 0             |
| f_5(x) = \sum_{i=1}^{n} 100(x_{i+1} - x_i)^2 + (x_i - 1)^2 | 30,60,100 | [-30,30]      | 0             |
| f_6(x) = \sum_{i=1}^{n} (x + 0.5)^2               | 30,60,100     | [-100,100]     | 0             |
| f_7(x) = \sum_{i=1}^{n} x_i^4 + \text{random}[0,1]   | 30,60,100     | [-128,128]     | 0             |

Table 4 The multimodal benchmark functions

| Function | Dimensions | Range          | Global optima |
|----------|------------|----------------|---------------|
| f_8(x) = 418.9829n + \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|}) | 30,60,100    | [-500,500]    | 0             |
| f_9(x) = \sum_{i=1}^{n} x_i^4 - 10 \cos(2\pi x_i) + 100 | 30,60,100 | [-5.12,5.12] | 0             |
| f_{10}(x) = e + 20 - 20 \exp\left(-0.2\sqrt{\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{2}{n} \sum_{i=1}^{n} \cos(2\pi x_i)\right) | 30,60,100 | [-32,32]      | 0             |
| f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \left(\frac{x_i}{\sqrt{10000}}\right) + 1 | 30,60,100 | [-600,600]    | 0             |
| y_i = 1 + \frac{x_i^{\alpha}}{\alpha}, u(x_a, a, b, c) = \begin{cases} b(x_i - a)^c & \text{if } a < x_i < b, \\ b(-x_i - a)^c & \text{if } x_i < -a \end{cases} | 30,60,100 | [-50,50]      | 0             |
\[ f_{12}(x) = \frac{n}{n-1} \left[ 10 \sin(\pi x_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[ 1 + 10 \sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 \right] + \sum_{i=1}^{n} u(x_i; 10, 100, 4) \]

\[ f_{13}(x) = 0.1 \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 \left[ 1 + \sin^2(3\pi x_1 + 1) \right] + (x_n - 1)^2 \left[ 1 + \sin^2(2\pi x_n) \right] + \sum_{i=1}^{n} u(x_i; 5, 100, 4) \]

| Function | Norm | TLBO  | STLBO1 | STLBO2 | STLBO3 | STLBO4 | STLBO5 | STLBO6 | STLBO7 |
|----------|------|-------|--------|--------|--------|--------|--------|--------|--------|
| F1       | Mean | 1.36E–236 | 1.34E–265 | 0 | 4.23E–222 | 0 | 1.46E–251 | 0 | 0 |
|          | Std  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|          | Rank | 4 | 2 | 5 | 1 | 3 | 1 | 1 | 1 |
|          | Mean | 3.00E–117 | 8.13E–130 | 1.72E–196 | 2.89E–109 | 6.29E–267 | 7.15E–125 | 7.16E–192 | 3.89E–250 |
| F2       | Mean | 1.12E–116 | 4.31E–129 | 0 | 7.90E–109 | 0 | 1.14E–124 | 0 | 0 |
|          | Std  | 5 | 4 | 8 | 1 | 6 | 3 | 2 | 0 |
|          | Mean | 4.10E–238 | 5.42E–265 | 0 | 1.77E–214 | 0 | 3.77E–245 | 0 | 0 |
| F3       | Mean | 8.72E–112 | 1.63E–125 | 3.90E–192 | 2.87E–102 | 7.14E–266 | 1.07E–118 | 1.06E–172 | 2.02E–246 |
|          | Std  | 3.06E–111 | 7.03E–125 | 0 | 1.20E–101 | 0 | 3.86E–118 | 0 | 0 |
|          | Rank | 7 | 5 | 3 | 8 | 1 | 6 | 4 | 2 |
| F4       | Mean | 8.92E+00 | 8.93E+00 | 8.98E+00 | 2.88E+00 | 8.94E+00 | 3.78E+00 | 4.41E+00 | 5.52E+00 |
|          | Std  | 6.37E–02 | 3.98E–02 | 1.72E–02 | 5.35E–01 | 4.59E–02 | 6.13E–01 | 1.26E+00 | 8.30E–01 |
|          | Rank | 5 | 6 | 8 | 1 | 7 | 2 | 3 | 4 |
| F5       | Mean | 9.87E–01 | 1.07E+00 | 1.85E+00 | 2.24E–06 | 1.47E+00 | 3.08E–06 | 2.43E–06 | 2.30E–06 |
|          | Std  | 4.58E–01 | 4.89E–01 | 5.36E–01 | 8.34E–07 | 5.52E–01 | 9.02E–07 | 4.15E–06 | 5.97E–07 |
|          | Rank | 5 | 6 | 8 | 1 | 7 | 4 | 3 | 2 |
| F6       | Mean | 2.36E–04 | 1.74E–04 | 8.48E–05 | 2.09E–04 | 2.15E–05 | 1.61E–04 | 9.69E–05 | 7.98E–05 |
|          | Std  | 1.38E–04 | 1.16E–04 | 6.60E–05 | 1.28E–04 | 3.64E–05 | 9.11E–05 | 6.64E–05 | 6.42E–05 |
|          | Rank | 8 | 6 | 3 | 7 | 1 | 5 | 4 | 2 |
| F7       | Mean | 1.80E+03 | 1.59E+03 | 2.27E+03 | 4.37E+02 | 2.37E+03 | 3.84E+02 | 4.00E+02 | 3.90E+02 |
|          | Std  | 2.15E+02 | 2.97E+02 | 3.07E+02 | 1.57E+02 | 2.50E+02 | 2.12E+02 | 2.46E+02 | 1.71E+02 |
|          | Rank | 7 | 6 | 8 | 5 | 1 | 2 | 4 | 3 |
| F8       | Mean | 3.23E–15 | 3.11E–15 | 2.44E–16 | 3.58E–15 | 4.44E–16 | 3.11E–15 | 4.44E–16 | 4.44E–16 |
|          | Std  | 6.38E–16 | 9.17E–16 | 0 | 1.21E–15 | 0 | 1.30E–15 | 0 | 0 |
|          | Rank | 5 | 2 | 4 | 1 | 3 | 1 | 1 | 1 |
| F9       | Mean | 3.86E+00 | 2.76E+00 | -2.19E+00 | -3.06E+00 | -2.42E+00 | -3.06E+00 | -3.06E+00 | -3.06E+00 |
|          | Std  | 2.66E–01 | 2.24E–01 | 4.79E–01 | 2.04E–07 | 3.54E–01 | 3.39E–07 | 1.69E–07 | 4.90E–03 |
|          | Rank | 3 | 4 | 1 | 6 | 2 | 7 | 5 | 8 |
| F10      | Mean | 6.42E–01 | 5.75E–01 | 1.04E+00 | 3.08E–06 | 8.95E–01 | 6.33E–06 | 2.42E–06 | 4.12E–06 |
|          | Std  | 2.78E–01 | 2.68E–01 | 3.37E–01 | 8.52E–07 | 2.21E–01 | 2.35E–06 | 1.19E–06 | 2.52E–06 |
|          | Rank | 6 | 5 | 8 | 1 | 7 | 4 | 2 | 3 |
| Average  | Mean | 4.8462 | 3.9231 | 3.6923 | 4.0769 | 2.4615 | 3.6154 | 2.5385 | 2.3846 |
| Overall  | Rank | 8 | 6 | 5 | 7 | 2 | 4 | 3 | 1 |
Table 6: The numerical results of 30D benchmark functions on TLBO and seven STLBOs

| Function | Norm | TLBO  | STLBO1 | STLBO2 | STLBO3 | STLBO4 | STLBO5 | STLBO6 | STLBO7 |
|----------|------|-------|--------|--------|--------|--------|--------|--------|--------|
| Mean     | 6.01E–234 | 1.97E–264 | 0      | 9.90E–208 | 0     | 8.56E–243 | 0     | 0     | 0     |
| F1 Std   | 0    | 0     | 0      | 0      | 0      | 0      | 0      | 0      | 0      |
| Rank     | 4    | 2     | 1      | 5      | 1      | 3      | 1      | 1      | 1      |
| Mean     | 3.24E–119 | 3.33E–131 | 1.97E–195 | 2.89E–105 | 1.39E–266 | 1.29E–120 | 2.95E–177 | 7.33E–245 |
| F2 Std   | 7.18E–119 | 1.14E–130 | 0      | 1.09E–104 | 0     | 2.75E–120 | 0     | 0      | 0     |
| Rank     | 7    | 5     | 3      | 8      | 1      | 6      | 4      | 2      | 0      |
| Mean     | 1.71E–234 | 3.96E–264 | 0      | 1.58E–200 | 0     | 1.73E–231 | 0     | 0      | 0     |
| F3 Std   | 0    | 0     | 0      | 0      | 0      | 0      | 0      | 0      | 0      |
| Rank     | 3    | 2     | 1      | 5      | 1      | 4      | 1      | 1      | 1      |
| Mean     | 5.88E–114 | 6.21E–128 | 6.11E–201 | 6.10E–97 | 4.93E–265 | 1.16E–114 | 5.90E–154 | 5.49E–233 |
| F4 Std   | 1.44E–113 | 1.78E–127 | 0      | 3.20E–96 | 0     | 2.80E–114 | 3.18E–153 | 0      | 0      |
| Rank     | 7    | 5     | 3      | 8      | 1      | 6      | 4      | 2      | 0      |
| Mean     | 2.89E+01 | 2.89E+01 | 2.90E+01 | 2.41E+01 | 2.90E+01 | 2.52E+01 | 2.51E+01 | 2.57E+01 |
| F5 Std   | 3.37E–02 | 3.61E–02 | 1.75E–02 | 1.03E+00 | 2.46E–02 | 6.21E–01 | 1.25E+00 | 1.07E+00 |
| Rank     | 5    | 6     | 7      | 1      | 8      | 3      | 2      | 4      | 0      |
| Mean     | 6.21E+00 | 5.97E+00 | 6.99E+00 | 5.86E–02 | 6.57E+00 | 3.45E–02 | 1.17E+01 | 9.17E+02 |
| F6 Std   | 8.05E–01 | 9.63E–01 | 6.46E–01 | 1.06E–01 | 5.89E–01 | 1.05E–01 | 1.67E–01 | 1.20E–01 |
| Rank     | 6    | 5     | 8      | 2      | 7      | 1      | 4      | 3      | 0      |
| Mean     | 2.05E–04 | 1.60E–04 | 6.90E–05 | 2.67E–04 | 6.84E–05 | 2.09E–04 | 9.06E–05 | 5.43E–05 |
| F7 Std   | 1.20E–04 | 1.33E–04 | 7.03E–05 | 1.66E–04 | 4.72E–05 | 1.33E–04 | 7.75E–05 | 5.42E–05 |
| Rank     | 8    | 6     | 3      | 7      | 2      | 5      | 4      | 1      | 0      |
| Mean     | 8.05E+03 | 8.21E+03 | 9.07E+03 | 3.43E+03 | 9.29E+03 | 3.83E+03 | 2.58E+03 | 3.02E+03 |
| F8 Std   | 5.62E+02 | 4.51E+02 | 5.01E+02 | 5.33E+02 | 4.66E+02 | 5.58E+02 | 4.20E+02 | 3.87E+02 |
| Rank     | 5    | 6     | 7      | 3      | 8      | 4      | 1      | 2      | 0      |
| Mean     | 0    | 0     | 0      | 0      | 0      | 0      | 0      | 0      | 0      |
| F9 Std   | 0    | 0     | 0      | 0      | 0      | 0      | 0      | 0      | 0      |
| Rank     | 1    | 1     | 1      | 1      | 1      | 1      | 1      | 1      | 1      |
| Mean     | 2.87E–15 | 3.11E–15 | -4.44E–16 | 4.29E–15 | -4.44E–16 | 3.70E–15 | -4.44E–16 | -4.44E–16 |
| F10 Std  | 1.28E–15 | 0      | 0      | 1.68E–15 | 0     | 1.32E–15 | 0     | 0      | 0      |
| Rank     | 2    | 3     | 1      | 5      | 1      | 4      | 1      | 1      | 0      |
| Mean     | 0    | 0     | 0      | 0      | 0      | 0      | 0      | 0      | 0      |
| F11 Std  | 0    | 0     | 0      | 0      | 0      | 0      | 0      | 0      | 0      |
| Rank     | 1    | 1     | 1      | 1      | 1      | 1      | 1      | 1      | 1      |
| Mean     | -2.47E–01 | -2.45E+00 | 5.37E–02 | -1.02E+00 | -3.03E–02 | -1.02E+00 | -1.02E+00 | -1.02E+00 |
| F12 Std  | 2.07E–01 | 2.06E–01 | 2.34E–01 | 2.42E–03 | 1.78E–01 | 4.13E–03 | 4.51E–03 | 6.09E–03 |
| Rank     | 3    | 8     | 2      | 4      | 1      | 5      | 6      | 7      | 0      |
| Mean     | 3.10E+00 | 3.52E+00 | 3.97E+00 | 5.25E–04 | 3.63E+00 | 4.10E–02 | 1.87E–02 | 2.14E–04 |
| F13 Std  | 5.20E–01 | 3.09E–01 | 4.30E–01 | 1.22E–03 | 3.46E–01 | 8.27E–02 | 5.13E–02 | 7.91E–05 |
| Rank     | 5    | 6     | 8      | 2      | 7      | 4      | 3      | 1      | 0      |
| Average  | 4.3846 | 4.3077 | 3.5385 | 4      | 3.0769 | 3.6154 | 2.5385 | 2.0769 | 0      |
| Overall  | 8    | 7     | 4      | 6      | 3      | 5      | 2      | 1      | 0      |

Table 7: The total ranking table

| Norm      | TLBO | STLBO1 | STLBO2 | STLBO3 | STLBO4 | STLBO5 | STLBO6 | STLBO7 |
|-----------|------|--------|--------|--------|--------|--------|--------|--------|
| Total Average Rank | 4.6154 | 4.1154 | 3.6154 | 4.0385 | 2.7692 | 3.6154 | 2.5385 | 2.2308 |
| Total Overall Rank  | 7    | 6      | 4      | 5      | 3      | 4      | 2      | 1      |
## Table 8 The numerical results of 40D benchmark functions on STLBO7 and other comparison algorithms

| Function | Norm | HS     | PSO    | MFO    | GA     | TLBO   | STLBO7 |
|----------|------|--------|--------|--------|--------|--------|--------|
|          |      | Mean   | Std    | Mean   | Std    | Mean   | Std    |
| F1       |      | 1.43E+02 | 2.64E+01 | 5.33E+03 | 2.81E+01 | 1.76E–233 | 0 |
| Rank     | 5    | 3      | 6      | 4      | 2      | 1      |
| Mean     | 3.25E+00 | 5.44E+16 | 8.94E+01 | 2.36E+00 | 3.73E–118 | 1.13E–241 |
| F2       |      | 5.59E–01 | 1.60E+17 | 2.65E+01 | 3.76E–01 | 1.26E–117 | 0 |
| Rank     | 4    | 6      | 5      | 3      | 2      | 1      |
| Mean     | 1.57E+04 | 5.08E+04 | 5.98E+04 | 2.55E+04 | 2.28E–236 | 0 |
| F3       |      | 3.53E+03 | 5.68E+04 | 2.11E+04 | 5.29E+03 | 0      | 0 |
| Rank     | 3    | 5      | 6      | 4      | 2      | 1      |
| Mean     | 1.61E+01 | 1.11E+01 | 6.22E+01 | 1.80E+01 | 2.10E–113 | 2.74E–234 |
| F4       |      | 1.98E+00 | 2.81E+00 | 1.11E+01 | 2.75E+00 | 8.15E–113 | 0 |
| Rank     | 4    | 3      | 6      | 5      | 2      | 1      |
| Mean     | 3.73E+03 | 7.11E+07 | 1.87E+07 | 1.12E+03 | 3.89E+01 | 3.56E+01 |
| F5       |      | 1.47E+03 | 7.26E+07 | 3.96E+07 | 6.88E+02 | 2.80E–02 | 9.37E–01 |
| Rank     | 4    | 6      | 5      | 3      | 2      | 1      |
| Mean     | 1.35E+02 | 3.51E+02 | 7.34E+03 | 2.74E+01 | 8.87E+00 | 2.31E–01 |
| F6       |      | 2.97E+01 | 1.81E+03 | 8.17E+03 | 7.84E+00 | 6.36E–01 | 1.84E–01 |
| Rank     | 5    | 4      | 6      | 3      | 2      | 1      |
| Mean     | 1.14E+05 | 1.47E+04 | 1.84E+09 | 5.72E+03 | 2.33E–04 | 6.96E–05 |
| F7       |      | 6.08E+04 | 2.95E+04 | 2.27E+09 | 6.49E+03 | 9.78E–05 | 6.70E–05 |
| Rank     | 5    | 4      | 6      | 3      | 2      | 1      |
| Mean     | 2.90E+02 | 8.77E+03 | 6.88E+03 | 1.08E+02 | 1.17E+04 | 4.99E+03 |
| F8       |      | 8.39E+01 | 1.58E+03 | 8.46E+02 | 4.25E+01 | 6.17E+02 | 6.77E+02 |
| Rank     | 2    | 5      | 4      | 1      | 6      | 3      |
| Mean     | 1.32E+01 | 4.98E+02 | 2.68E+02 | 1.19E+01 | 0      | 0      |
| F9       |      | 2.49E+00 | 1.40E+02 | 5.52E+01 | 2.68E+00 | 0      | 0      |
| Rank     | 3    | 5      | 4      | 2      | 1      | 1      |
| Mean     | 3.54E+00 | 1.75E+01 | 2.00E+01 | 2.61E+00 | 3.35E–15 | -4.44E–16 |
| F10      |      | 3.98E–01 | 2.77E+00 | 1.11E–03 | 2.51E–01 | 8.86E–16 | 0 |
| Rank     | 4    | 5      | 6      | 3      | 2      | 1      |
| Mean     | 2.22E+00 | 3.03E–01 | 6.66E+00 | 1.24E+00 | 0      | 0      |
| F11      |      | 3.18E–01 | 2.61E–01 | 7.69E+01 | 7.09E–02 | 0      | 0      |
| Rank     | 4    | 2      | 5      | 3      | 1      | 1      |
| Mean     | 5.35E–02 | 1.71E+07 | 1.71E+07 | -1.38E–01 | 1.34E–01 | -7.60E–01 |
| F12      |      | 3.07E–01 | 6.39E+07 | 6.39E+07 | 7.39E–02 | 1.18E–01 | 5.85E–03 |
| Rank     | 1    | 5      | 5      | 3      | 2      | 4      |
| Mean     | 8.38E+00 | 5.47E+07 | 1.37E+07 | 9.91E+00 | 4.63E+00 | 5.88E–02 |
| F13      |      | 2.57E+00 | 1.39E+08 | 7.36E+07 | 2.95E–01 | 5.26E–01 | 9.37E–02 |
| Rank     | 4    | 6      | 5      | 2      | 3      | 1      |
| Mean     | 3.6923 | 4.5385 | 5.3077 | 3      | 2.3077 | 1.3846 |
| Average Rank | 4    | 5      | 6      | 3      | 2      | 1      |

## Table 9 The numerical results of 70D benchmark functions on STLBO7 and other comparison algorithms

| Function | Norm | HS     | PSO    | MFO    | GA     | TLBO   | STLBO7 |
|----------|------|--------|--------|--------|--------|--------|--------|
|          |      | Mean   | Std    | Mean   | Std    | Mean   | Std    |
| F1       |      | 5.28E+03 | 4.72E+03 | 1.94E+04 | 1.82E+02 | 7.59E–237 | 0 |
| Std      | 7.43E+02 | 2.18E+04 | 1.10E+04 | 5.50E+01 | 0      | 0      |
| Rank | 5 | 4 | 6 | 3 | 2 | 1 |
|------|---|---|---|---|---|---|
| Mean | 3.39E+01 | 1.60E+32 | 1.57E+02 | 8.43E+00 | 1.28E–118 | 3.93E–240 |
| F2   | Std 2.78E+00 | 5.50E+32 | 5.49E+01 | 9.22E–01 | 5.70E–118 | 0 |
| Rank | 4 | 6 | 5 | 3 | 2 | 1 |
| Mean | 9.66E+04 | 1.09E+05 | 1.81E+05 | 7.93E+04 | 2.71E–237 | 0 |
| F3   | Std 1.94E+04 | 1.46E+05 | 4.68E+04 | 1.05E+04 | 0 | 0 |
| Rank | 4 | 5 | 6 | 3 | 2 | 1 |
| Mean | 3.83E+01 | 3.43E+01 | 8.70E+01 | 4.06E+01 | 1.45E–113 | 1.84E–227 |
| F4   | Std 2.21E+00 | 1.16E+01 | 4.94E+00 | 4.74E+00 | 7.04E–113 | 0 |
| Rank | 4 | 3 | 6 | 5 | 2 | 1 |
| Mean | 1.59E+06 | 1.50E+08 | 4.07E+07 | 5.10E+03 | 6.89E+01 | 6.60E+01 |
| F5   | Std 4.49E+05 | 8.32E+07 | 6.42E+07 | 1.57E+03 | 2.62E–02 | 1.03E+00 |
| Rank | 4 | 6 | 5 | 3 | 2 | 1 |
| Mean | 5.26E+03 | 6.43E+02 | 2.03E+04 | 1.87E+02 | 1.61E+01 | 9.92E–01 |
| F6   | Std 8.90E+02 | 1.80E+03 | 1.31E+04 | 5.09E+01 | 8.24E–01 | 4.88E–01 |
| Rank | 5 | 4 | 6 | 3 | 2 | 1 |
| Mean | 1.48E+08 | 4.53E+07 | 6.95E+09 | 2.38E+05 | 1.95E–04 | 9.01E–05 |
| F7   | Std 4.32E+07 | 2.41E+08 | 6.08E+09 | 2.01E+05 | 1.14E–04 | 8.28E–05 |
| Rank | 5 | 4 | 6 | 3 | 2 | 1 |
| Mean | 3.43E+03 | 1.53E+04 | 1.34E+04 | 6.51E+02 | 2.28E+04 | 1.20E+04 |
| F8   | Std 5.47E+02 | 2.46E+03 | 9.99E+02 | 1.45E+02 | 6.74E+02 | 1.71E+02 |
| Rank | 2 | 5 | 4 | 1 | 6 | 3 |
| Mean | 1.44E+02 | 9.02E+02 | 5.25E+02 | 4.19E+01 | 0 | 0 |
| F9   | Std 1.51E+01 | 2.37E+02 | 7.74E+01 | 4.29E+00 | 0 | 0 |
| Rank | 3 | 5 | 4 | 2 | 1 | 1 |
| Mean | 1.02E+01 | 1.89E+01 | 2.00E+01 | 3.89E+00 | 3.35E–15 | -4.44E–16 |
| F10  | Std 5.59E–01 | 5.02E–01 | 3.82E–04 | 3.05E–01 | 1.28E–15 | 0 |
| Rank | 4 | 5 | 6 | 3 | 2 | 1 |
| Mean | 4.77E+00 | 1.04E+00 | 1.43E+02 | 2.77E+00 | 0 | 0 |
| F11  | Std 4.23E+00 | 1.19E–01 | 1.09E+02 | 3.82E–01 | 0 | 0 |
| Rank | 4 | 2 | 5 | 3 | 1 | 1 |
| Mean | 1.46E+04 | 1.62E+08 | 7.68E+07 | 9.54E–02 | 5.93E–01 | -4.20E–01 |
| F12  | Std 1.79E+04 | 2.04E+08 | 1.35E+08 | 1.70E–01 | 1.11E–01 | 2.33E–02 |
| Rank | 4 | 6 | 5 | 1 | 3 | 2 |
| Mean | 1.01E+06 | 3.78E+08 | 5.53E+07 | 7.13E+00 | 8.92E+00 | 3.61E–01 |
| F13  | Std 5.13E+05 | 4.53E+08 | 1.39E+08 | 1.42E+00 | 5.49E–01 | 2.22E–01 |
| Rank | 4 | 6 | 5 | 2 | 3 | 1 |
| Average Rank | 4 | 4.6923 | 5.3077 | 2.6923 | 2.3077 | 1.2308 |
| Overall Rank | 4 | 5 | 6 | 3 | 2 | 1 |

Table 10 The numerical results of 100D benchmark functions on STLBO7 and other comparison algorithms
|        | Mean   | Std    | Rank |        | Mean   | Std    | Rank |        | Mean   | Std    | Rank |        |
|--------|--------|--------|------|--------|--------|--------|------|--------|--------|--------|------|--------|
| F3     | 2.39E+05 | 3.13E+05 | 3.52E+05 | 1.73E+05 | 1.92E–233 | 0     |      |        |        |        |      |        |
|        | 2.73E+04 | 3.32E+05 | 8.05E+04 | 2.71E+04 | 0              | 0     |      |        |        |        |      |        |
|        | 4      | 5      | 6     | 3      | 2             | 1     |      |        |        |        |      |        |
|        | 5.26E+01 | 5.76E+01 | 9.40E+01 | 5.68E+01 | 8.16E–114     | 2.43E–226 |      |        |        |        |      |        |
| F4     | 2.44E+00 | 1.66E+01 | 2.98E+00 | 4.07E+00 | 3.58E–113     | 0     |      |        |        |        |      |        |
|        | 3      | 5      | 6     | 4      | 2             | 1     |      |        |        |        |      |        |
|        | 1.68E+07 | 1.97E+08 | 5.34E+07 | 2.45E+04 | 9.89E+01     | 9.64E+01 |      |        |        |        |      |        |
| F5     | 1.85E+06 | 1.44E+08 | 5.61E+07 | 7.47E+03 | 3.14E–02     | 1.12E+00 |      |        |        |        |      |        |
|        | 4      | 6      | 5     | 3      | 2             | 1     |      |        |        |        |      |        |
|        | 2.10E+04 | 4.36E+03 | 3.38E+04 | 7.03E+02 | 2.34E+01     | 2.77E+00 |      |        |        |        |      |        |
| F6     | 2.25E+03 | 4.65E+03 | 1.67E+04 | 1.40E+02 | 7.74E–01     | 8.76E–01 |      |        |        |        |      |        |
|        | 5      | 3      | 6     | 4      | 2             | 1     |      |        |        |        |      |        |
|        | 2.27E+09 | 2.28E+08 | 1.77E+10 | 2.56E+06 | 1.95E–04     | 8.04E–05 |      |        |        |        |      |        |
| F7     | 3.51E+08 | 6.87E+08 | 1.15E+10 | 1.18E+06 | 1.14E–04     | 5.48E–05 |      |        |        |        |      |        |
|        | 5      | 4      | 6     | 3      | 2             | 1     |      |        |        |        |      |        |
|        | 7.55E+03 | 2.25E+04 | 2.10E+04 | 1.96E+03 | 3.35E+04     | 1.96E+04 |      |        |        |        |      |        |
| F8     | 6.15E+02 | 5.41E+03 | 1.18E+03 | 3.51E+02 | 8.31E+02     | 1.23E+03 |      |        |        |        |      |        |
|        | 2      | 5      | 4     | 1      | 6             | 3     |      |        |        |        |      |        |
|        | 3.17E+02 | 1.15E+03 | 8.05E+02 | 8.96E+01 | 0              | 0     |      |        |        |        |      |        |
| F9     | 2.17E+01 | 2.94E+02 | 8.29E+01 | 7.62E+00 | 0              | 0     |      |        |        |        |      |        |
|        | 5      | 4      | 6     | 3      | 2             | 1     |      |        |        |        |      |        |
|        | 7.55E+03 | 2.25E+04 | 2.10E+04 | 1.96E+03 | 3.35E+04     | 1.96E+04 |      |        |        |        |      |        |
| F10    | 3.61E–01 | 2.75E–01 | 1.18E–05 | 3.31E–01 | 0              | 0     |      |        |        |        |      |        |
|        | 4      | 5      | 6     | 3      | 2             | 1     |      |        |        |        |      |        |
|        | 1.84E+02 | 1.33E+00 | 3.22E+02 | 6.68E+00 | 0              | 0     |      |        |        |        |      |        |
| F11    | 1.78E+01 | 1.38E–01 | 1.48E+02 | 1.17E+00 | 0              | 0     |      |        |        |        |      |        |
|        | 4      | 2      | 5     | 3      | 1             | 1     |      |        |        |        |      |        |
|        | 5.02E+06 | 3.59E+08 | 7.78E+07 | 1.24E+00 | 7.36E–01     | 2.76E–01 |      |        |        |        |      |        |
| F12    | 1.76E+06 | 2.35E+08 | 1.29E+08 | 3.53E–01 | 8.27E–02     | 1.40E–02 |      |        |        |        |      |        |
|        | 4      | 6      | 5     | 1      | 3             | 2     |      |        |        |        |      |        |
|        | 3.55E+07 | 8.22E+08 | 3.18E+08 | 2.38E+01 | 1.34E+01     | 1.03E+00 |      |        |        |        |      |        |
| F13    | 6.37E+06 | 5.19E+08 | 3.46E+08 | 5.52E+00 | 4.38E–01     | 4.22E–01 |      |        |        |        |      |        |
|        | 4      | 6      | 5     | 3      | 2             | 1     |      |        |        |        |      |        |
|        | 3.9231 | 4.7692 | 5.3077 | 2.7692 | 2.2308 | 1.2308 |      |        |        |        |      |        |
|        | 4      | 5      | 6     | 3      | 2             | 1     |      |        |        |        |      |        |

Table 11: The total ranking table

| Norm | HS   | PSO  | MFO  | GA   | TLBO | STLBO |
|------|------|------|------|------|------|-------|
| Total Average Rank | 3.8718 | 4.6667 | 5.3077 | 2.8205 | 2.2051 | **1.2821** |
| Total Overall Rank | 4 | 5 | 6 | 3 | 2 | **1** |
Fig. 4. Typical convergence graph of six different algorithms for functions 1 to 13 ($D = 70$).
(a) Function F1; (b) Function F2; (c) Function F3; (d) Function F4; (e) Function F5; (f) Function F6; (g) Function F7.
5 Conclusion and future research

In this paper, a novel strengthened TLBO called STLBO based on three enhancement mechanisms is proposed to improve the search performance of the traditional TLBO. See Section 3 for details, these three strengthening mechanisms are a linear increasing teaching factor for more realistic and faster convergence, an elite system based on a new teacher and a class leader to strengthen the value of individual elites and promote the evolution of the class in the right direction, and a Cauchy mutation mechanism to increase population diversity and improve algorithm exploration property. According to the different introduction forms of the three strengthening mechanisms, seven variants of STLBO are designed. Subsequently, two comparative experiments based on thirteen famous benchmark functions are implemented to verify the search performance of the proposed STLBOs. In Section 4.2, the numerical results between TLBO and STLBOs on 13 functions with 10D and 30D are recorded, compared and analyzed. The results reveal that the seven STLBOs evidently improved the TLBO performance. Among, STLBO7 defeated other competitors and ranked first, while the original TLBO ranked last. In Section 4.2, a set of comparative experiments is implemented with STLBO7 and several advanced optimizers such as HS, PSO, MFO, and GA to further measure the effectiveness of the new algorithm. The experimental results on 13 functions with 40D, 70D, and 100D prove that STLBO7 is the most excellent optimization method. STLBO7 exhibits extraordinary performance on most problems compared with the other optimization algorithms. Further, in order to more intui-

Fig. 5. Typical convergence graph of six different algorithms for functions 8 to 13 (D = 70).
(a) Function F8; (b) Function F9; (c) Function F10; (d) Function F11; (e) Function F12; (f) Function F13.
tively explain the algorithm performance of STLBO7, convergence curve of relevant experiments is provided. It can be seen from the convergence curve that STLBO7 has a higher solution accuracy, faster convergence speed and better avoidance of local optima.

In future research, the STLBOs will be applied to more practical tasks and multi-target issues. Meanwhile, designing a binary version of STLBOs and applying them to practical problems is also a worthwhile research direction.

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Funding This work was supported by the National Natural Science Foundation of China (No.71840003), Key Soft Science Project of Science and Technology Innovation Action Plan of Shanghai Municipal Science and Technology Commission (No. 20692104300), and Science and Technology Development Foundation of University of Shanghai for Science and Technology (No.2018KJFZ043).

Conflict of interest The authors declare that they have no conflict of interest.

Author Contributions conceptualization, Xuefen Chen, Chunming Ye and Yang Zhang; methodology, Xuefen Chen; software, Xuefen Chen; validation, Chunming Ye; formal analysis, Xuefen Chen; investigation, Lingwei Zhao, Jing Guo, Kun Ma; writing—original draft preparation, Xuefen Chen; writing—review and editing, Xuefen Chen; visualization, Xuefen Chen; supervision, Chunming Ye.

References

[1] Bidar M, Kanan HR, Mouhoub M, Sadaoui S (2018) Mushroom Reproduction Optimization (MRO): A Novel Nature-Inspired Evolutionary Algorithm. In: 2018 IEEE Congress on Evolutionary Computation (CEC) pp 1–10
[2] Zhang Y, Zhou XZ, Shih PC (2020) Modified Harris hawks optimization algorithm for global optimization problems. Arab J Sci Eng 45(12):10949–10974
[3] Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495-513
[4] Deb K, Pratap A, Agarwal S, et al (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182-197
[5] Marini F, Walczak B (2015) Particle swarm optimization (PSO). A tutorial. Chemometr Intell Lab 149:153-165
[6] Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
[7] Alsewari AA, Kabir MN, Zamli KZ, et al (2019) Software Product Line Test List Generation based on Harmony Search Algorithm with Constrant Ant colony optimization nts Support. Inter-national J Advanced Comput Sci Appl 10(1):605–610
[8] Lee KS, Geem ZW (2005) new meta-heuristic algorithm for con-tinuous engineering optimization: harmony search theory and practice. Comput Method Appl M 194(36-38):3902–3933
[9] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: Algorithm and applications. Future Gener Comp Sy 97: 849–872
[10] Jain M, Singh V, Rani A (2019) A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm Evol Comput 44:148–175
[11] Wang GG, Deb S, Gao XZ, et al (2016) A new metaheuristic optimisation algorithm motivated by elephant herding behavior. Int J Bio-Inspir Com 8(6):394–409
[12] Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: A Grav-itational Search Algorithm. Inform Sciences 179:2232–2248
[13] Wang GG, Deb S, Cui ZH (2019) Monarch butterfly optimization. Neural Comput Appl 31(7):1995–2014
[14] Wang GG, Deb S, Coelho LDS (2018) Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems. Int J Bio-Inspir Com 12(1): 1–22
[15] mirjalili S, Gandomi AH, Mirjalili SZ, et al (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
[16] Zhang HZ, Liu F, Zhou YY, et al (2020) A hybrid method integrating an elite genetic algorithm with tabu search for the quad-ratic assignment problem. Infrom Sciences 539:347–374
[17] Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun. Nonlinear Sci 17(12):4831–4845
[18] Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical de-sign optimization problems. Comput Aided Design 43(3): 303–315
[19] Reddy MC, Rao KV, Suresh G (2021) An experimental investigation and optimization of energy consumption and surface defects in wire cut electric discharge machining. J Alloy Compd 861:158582

[20] Sahoo AK, Mishra SK, Majhi B, et al (2021) Real-Time Identification of Fuzzy PID-Controlled Maglev System using TLBO-Based Functional Link Artificial Neural Network. Arab J Sci Eng 46(4): 4103–4118

[21] Sameer FO, Al-Obaidi MJ, Al-Bassam WW, et al (2021) Multi-objectives TLBO hybrid method to select the related risk features with rheumatism disease. Neural Comput Appl

[22] Rao RV, Patel V (2013) An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. Sci Iran 20(3):710–720

[23] Niu Q, Zhang HY, Li K (2014) An improved TLBO with elite strategy for parameters identification of PEM fuel cell and solar cell models. Int J Hydrogen Energ 39(8):3837–3854

[24] Sultana S, Roy PK (2014) Multi-objective quasi-oppositional teaching learning based optimization for optimal location of distributed generator in radial distribution systems. Int J Elec Power 63:534–545

[25] Jiang ZQ, Zou F, Chen DB, et al (2021) An improved teaching–learning-based optimization for multilevel thresholding image segmentation. Arab J Sci Eng

[26] Zhan ZH, Zhang J, Li Y, et al (2009) Adaptive Particle Swarm Optimization. IEEE T Syst Man Cy B 39(6):1362–1381

[27] Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE T Evolut Comput 3:82–102

[28] Digalakis JG, Aaritis KG (2001) On benchmarking functions for genetic algorithms. Int J Comput Math 77:81–506