Intelligent Systems with Applications 17 (2023) 200163

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

An advanced teaching-learning-based algorithm to solve unconstrained optimization problems

Mohammad Fatehi [a], Alireza Toloei [a], S.T.A. Niaki [b,*], Enrico Zio [c,d]

[a] Department of Aerospace Engineering, Shahid Beheshti University, Tehran, Iran
[b] Department of Industrial Engineering, Sharif University of Technology, Iran
[c] MINES Paris-PSL University, France
[d] Energy Department, Politecnico di Milano, Italy

A R T I C L E I N F O

Keywords:
Intelligent optimization algorithms
Teaching-learning-based optimization (TLBO)
Advanced teaching-learning-based optimization (Ad-TLBO)

A B S T R A C T

The Teaching-Learning-Based Optimization (TLBO) algorithm is being extended to a broader range of applied optimization problems in the literature, mimicking the teaching-learning process. This paper proposes an Advanced Teaching-Learning-Based Optimization (Ad-TLBO) algorithm to enhance the efficiency and performance of the original version of TLBO in terms of accuracy, convergence rate, and reliability characteristics. The advancement is obtained by modifying the initialization, search approach, and structure of the two main phases of this algorithm in four steps to improve exploration and exploitation capability. Efficiency comparisons are shown in four challenges with various benchmark functions with multimodal, separable, differentiable, and continuity characteristics. The results are compared with several intelligent optimization algorithms. It is also deduced that this algorithm outperforms all investigated optimization algorithms in terms of accuracy, convergence speed, and success to reach acceptable solutions for various benchmark functions.

Introduction

Many advanced optimization algorithms have been proposed to solve optimization problems so far. These algorithms are classified into different categories. Population-based heuristic algorithms are one of the essential groups among these categories. Genetic Algorithms (GA), Artificial Bee Colony (ABC), and Particle Swarm are some recognized algorithms in this group. Most of these algorithms require specific control parameters. For instance, mutation probability, selection operator, and crossover probability are particular control parameters for the Genetic Algorithm. There are no general guidelines on tuning these parameters, while the values of these control parameters directly affect the algorithm performance. This issue led to the attraction of researchers to develop algorithms with less specific parameters.

In 2011 Rao et al. (2011) introduced an optimization algorithm called Teaching-learning-based optimization (TLBO) that only requires common control parameters such as initial population size, maximum iteration, and so on as the input (Rao & Patel, 2013). It is inspired by the teaching-learning process and falls within the population-based heuristic stochastic optimization algorithms (Zou et al., 2019).

Some characteristics such as simple concepts, no requirements to specific parameters, rapid convergence, easy implementation, and effectiveness have led to the widespread use of this algorithm in various fields, even in recent years. Since introducing the TLBO method until now, many researchers have tried to improve the algorithm's performance with many ideas. Usually, these improvement ideas either lead to changing the structure and basic concepts of the algorithm, such as search and initialization techniques, adaptive parameters, and modifying the structure of the two phases of teaching and learning, or lead to combining this algorithm with other optimization algorithms or search techniques and creating hybrid methods (Zou et al., 2019). In the following, prominent examples of each type of change are mentioned. Rao (2016) provided a list of 200 papers on the application of this algorithm. Moreover, a survey paper was published in 2019 (Zou et al., 2019) that more comprehensively studied this algorithm's modification and engineering applications. In Shao et al. (2017) proposed an initialization technique of combining a modified Nawaz-Enscore-Ham (NEH) heuristic and the opposition-based learning (OBL) to generate the initial population. To improve the exploration and search capability, convergence, and maintain the diversity of the population, many researchers, such as Ji et al. (2017), have used genetic algorithm operators such as the mutation and the crossover operators into TLBO. Bureerat

* Corresponding author.
E-mail address: Niaki@Sharif.edu (S.T.A. Niaki).

https://doi.org/10.1016/j.iswa.2022.200163
Received 3 January 2022; Received in revised form 4 October 2022; Accepted 1 December 2022
Available online 5 December 2022

et al. in 2021 (Bureerat & Sleesongsom, 2021) proposed a new self-adaptive TLBO with a diversity archive (ATLBO-DA) and employed it for the four-bar linkage path generation problem. Chen et al. (2016b) proposed the VTLBO method, which uses a variable population scheme in a triangle form. Decreasing the computational cost and extending it for parameters optimization of artificial neural network (ANN) are the objects of this proposed method. Algorithms (TLBO-MA) (Qu et al., 2017), (HTLBO) (Tang et al., 2017), and Das & Padhy (2018) can be mentioned as prominent examples of developed hybrid TLBO algorithms with search techniques. Qu et al. (2017) proposed a novel TLBO memetic algorithm (TLBO-MA) that combined TLBO and Multi-meme learning based on meta-Lamarckian. Tang et al. (2017) proposed a hybrid TLBO (HTLBO) approach by combining TLBO, VNS[1] and crossover operator. Das & Padhy (2018) developed a novel hybrid model using TLBO and SVR[2] (Zou et al., 2019). Refs. (Chen et al. 2018, Dib & Boumhidi 2017, Patel et al. 2017) are examples of the hybrid method of TLBO with other optimization methods, which are combined with Differential Evolution (DE), Artificial Bee Colony (ABC), and Particle Swarm Optimization (PSO) methods, respectively. Even recently, investigation on this method continues in papers such as (Ahmadi-Nedushan & Fathnejat, 2022; Bui et al., 2022; Kumar et al., 2022; Ren et al., 2022; Wu et al., 2022). The proposed method has been compared with some of these mentioned methods, such as original TLBO, ETLBO, sawTLBO (Rao & Patel, 2012), VTLBO (Chen et al., 2016b), and ATLBO-DA (Bureerat & Sleesongsom, 2021), in the second challenge. As shown in Tables 5–8, the proposed method outperformed the other methods for most benchmark functions.

Although researchers have many studies on the development of the original TLBO method and have obtained acceptable results, none of the versions of this method have been able to outperform the leading optimization algorithms that have been ranked in competitions such as the CEC competition. This paper proposes an advanced version by combining improvement in the search and initialization techniques and the main concept of the algorithm in two phases of teaching and learning, leading to a method that can have more acceptable results compared to the leading optimization algorithms in competitions like CEC. The proposed method's results are compared to the CEC competition's best-ranked methods and other recently advanced optimization algorithms in the third challenge.

Improving algorithm performance in terms of initialization, convergence rate, and exploration and exploitation characteristics are the main goals of the proposed method. This advancement is performed in two general parts. The first part is the internal restructuring of the initial TLBO algorithm, which is done in four steps. The second part is related to the search method in the possible space and includes two techniques of initialization and searching. This part is not limited to the TLBO method and can be applied to other methods, but the convergence speed of the TLBO method makes this part more efficient. Although this advanced method has more control parameters than the original method, these changes significantly increase the approach efficiency and quality of answers.

The article structure is expressed as follows: After this introduction, Section 2 provides a brief description of the original TLBO. The details of the advanced version of TLBO are explained in section 3. In Section 4, four challenges are used to investigate the performance of the proposed algorithm. The proposed algorithm results are compared with some of the most widely used intelligent optimization algorithms in these challenges. Subsequently, in Section 5, the results are discussed, and we conclude the work by summarizing the performance of the proposed version.

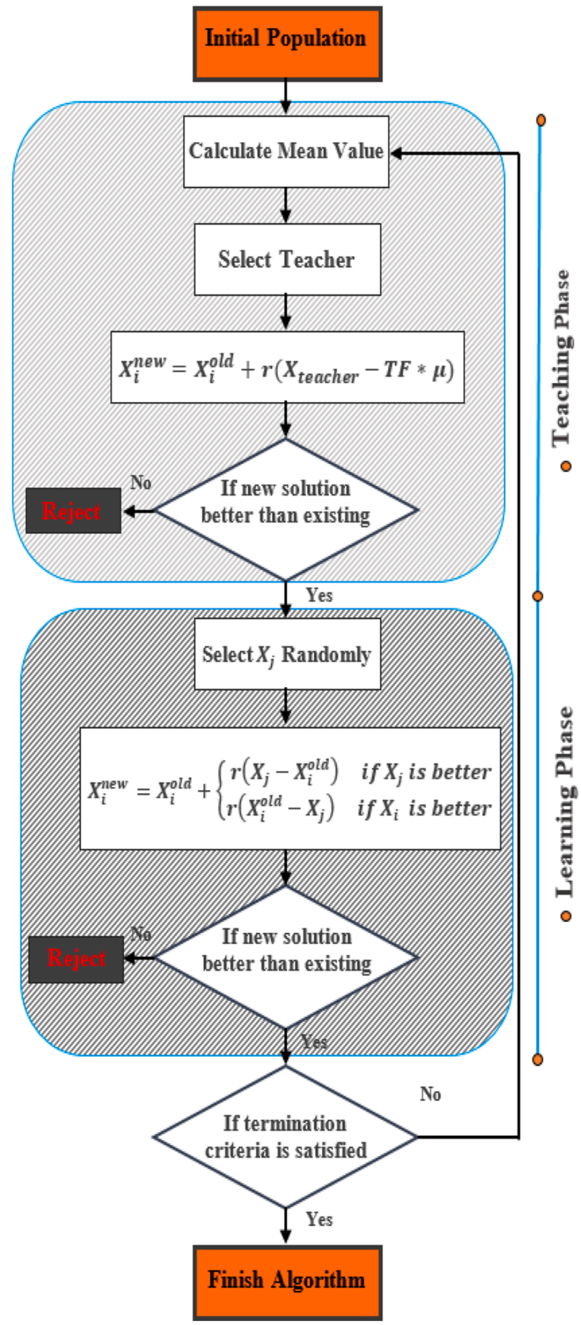

Fig. 1. TLBO algorithm framework.

Teaching-learning based optimization

The TLBO algorithm is based on increasing the knowledge of the class population by the Teacher's influence on the learners and learners' influence on each other. Hence the structure of this algorithm has two main parts. The first part is the teaching phase, which expresses the Teacher's interaction with other members. The second part is the learning phase, which represents the interaction of members with each other. The process of this algorithm is as follows: in the first step, an initial population is generated randomly, and then the mean value of this population is calculated. The best member of the population (the population member with a minimum value at minimization problem) is selected as a teacher, and the other members are known as the learners. After performing the teaching and learning phases, if the termination criteria are satisfied, the algorithm will end; otherwise, the algorithm

[1] Variable Neighborhood Search.
[2] Support Vector Machine.

process should be repeated with the new population generated after these two phases. Fig. 1 shows the framework of this algorithm. Refer to Refs. (Rao, 2016; Rao & Patel, 2013; Rao et al., 2011; Zou et al., 2019) for more detailed information about this algorithm. The teaching and learning phases are briefly described below.

Teaching phase

The teaching phase is the first part of the TLBO algorithm in which the Teacher tries to bring the other member's knowledge closer to his (her) knowledge level. In mathematics format, this issue is expressed by attracting other members' mean fitness value closer to the best answer (Teacher). This phase is formulated as Eq. (1).

$$X_i^{new} = X_i^{old} + r(X_{teacher} - TF * \mu), \tag{1}$$

where μ is the mean value of the population. X_i^{old} and X_i^{new} are the values of the i th member of the population before and after the teaching phase, respectively. Besides, r is a uniform random number in the interval [0, 1], and TF is the teaching factor that decides the mean value to be changed. Rao et al. (2011) concluded with experience on several benchmarks that this value should be either 1 or 2.

Learning phase

The second part of this algorithm is the learning phase. In this phase, a learner interacts randomly with other learners to increase his (or her) knowledge. If his (or her) level is higher than the randomly selected learner, he (or she) gets away from the random learner, and else, he (or she) approaches the other learner. The formulation of this phase is given in the following equation.

$$X_i^{new} = X_i^{old} + \begin{cases} r\left(X_j - X_i^{old}\right) & if\ X_j\ is\ better \\ r\left(X_i^{old} - X_j\right) & if\ X_i\ is\ better \end{cases}, \tag{2}$$

where X_j is the randomly selected learner to interact with X_i.

Advanced teaching-learning based optimization (Ad-TLBO)

In this section, an advanced version of the TLBO method is proposed. This advancement is made in two general parts to enhance the performance of the original TLBO in terms of accuracy, convergence rate, and reliability characteristics. In the first part, the structure of the teaching and learning phases is modified in four steps. In the second part, the generation of the initial population and the exploration approach of the original TLBO is improved with a kind of search space segmentation.

Part I

The structure modification of the teaching and learning phases is expressed in the following four steps.

Step1: The first step is to modify the structure of the learning phase. As mentioned, in this phase, learners interact with each other. Each learner interacts with another learner randomly selected from the population in this procedure. Due to the random selection of the second learner, it is always possible to obtain a better result by selecting and interacting with another learner. With that in mind, one idea for choosing the best learner is to interact with all the learners and then choose the best outcome. This approach is proposed in Ref. Rao (2016). This reference shows that although the number of function evaluations of this approach increases, the optimal global solution is obtained in fewer iterations. However, this approach has the disadvantage of increasing the number of function evaluations.

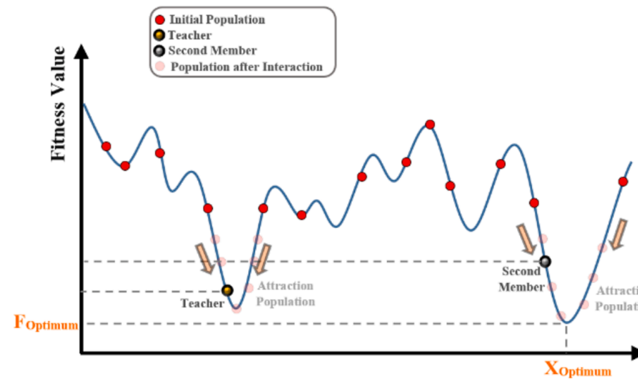

Fig. 2. Alternative candidate strategy.

A new approach is proposed in this step to overcome this problem. We used the approach of Ref. Rao (2016) for several benchmarks. In the set of interactions, it was observed that for almost all learners, interaction with the Teacher is better than interaction with other learners. Therefore, this step proposes that instead of random interactions or interactions with all learners, all learners interact only with the Teacher. Comparing the proposed approach results with Ref. Rao (2016) in many benchmarks shows that both approaches have similar results. However, in addition to keeping the advantage of low iteration numbers, the proposed approach also has the advantage of a significantly lower number of function evaluations (i.e., equal to the number of population members (n)). In the approach of Ref. Rao (2016), all members interact with each other; therefore, the number of function evaluations increases to the number of populations members to the power of two (i.e., $n * n$). The formulation of the learning phase (Eq. (2)) is modified as follows.

$$X_i^{new} = X_i^{old} + r\left(X_{Teacher} - X_i^{old}\right), \tag{3}$$

where $X_{Teacher}$ is the Teacher's position to interact with X_i. In the original method, the Teacher is selected before the teaching phase. According to the modification of this step, choosing a new teacher after the teaching phase can improve the performance of these changes in the learning phase because teacher value after the teaching phase is better than before. Therefore, modifying Eqs. (3) to (4) improves the results.

$$X_i^{new} = X_i^{old} + r\left(X_{Teacher}^{New} - X_i^{old}\right), \tag{4}$$

Fig. 5 shows the performance of the first step compared to the original method on the sphere benchmark. Diagram step1$'$ is the result of the teacher selection before the teaching phase, and the diagram step1 is the result of the teacher selection after the teaching phase.

Step2: As shown in Fig. 1, there are two selection operators in the teaching and learning phases of the original version of the TLBO method. In these operators, a binary comparison is made between the statuses of each member before and after the interaction (i.e., a comparison between X_i^{old} and X_i^{new}). The better fitness value is entered into the new population. In this approach, one of the statuses of each variable must be eliminated. However, the eliminated status of this variable may have a better result than the two statuses of the pre-and post-interaction of another variable. This forced elimination may reduce the convergence rate of the algorithm.

In this step, it is proposed that for the two phases, all members of the population in both statuses of pre-and post-interaction are gathered in a new population with $2n$ members (n is the population number). After the fitness value-based sort, the first n members enter the new final population. The optimization process continues with the new population obtained. As shown in Fig. 5, this step increases the accuracy and the

Table 1
Six rank intervals for a fitness value.

$Rank_f$	Bounds
1	$F_{teacher} + [0, \sigma_F]$
2	$F_{teacher} + [\sigma_F, 2\sigma_F]$
3	$F_{teacher} + [2\sigma_F, 3\sigma_F]$
4	$F_{teacher} + [3\sigma_F, 4\sigma_F]$
5	$F_{teacher} + [4\sigma_F, 5\sigma_F]$
6	$F_{teacher} + [5\sigma_F, 6\sigma_F]$

convergence rate of the original TLBO because valuable population members are not necessarily eliminated in the binary comparison. Note that although the algorithm's performance is improved in this step, no function evaluation is added to the original version.

Step3: This step proposes a more fundamental change than the previous steps. The Teacher has an essential and critical role in the TLBO method and its improved versions. Modifications of the present study have also increased the importance of the Teacher in the optimization process. The algorithm attracts learners to the teacher region during the optimization process. Therefore, improper teacher selection can reduce the performance of the optimization method and increase the probability of being trapped in the local optimal solutions. For example, if the Teacher is in the optimal local region, the optimization approach can even fail to reach the optimal global answer in some cases. Hence the exploration of the method must be improved. Fig. 2 shows the performance of an alternative candidate to decrease the probability of being trapped in the local optimal solutions.

On the other hand, as mentioned in step 1, by implementing the Ref. Rao (2016) approach, i.e., interaction with all members and selecting the best interaction, it was observed that interaction with the Teacher is the best interaction for almost all benchmarks. However, in some cases, it was observed that interaction with a member other than the Teacher and close to its value has the better interaction. In this step, an alternative candidate is proposed as a second person to avoid the disadvantages.

This candidate must have two characteristics. First, this member must be an acceptable fitness value as a candidate to attract other learners to the optimal global region. Given that the Teacher is the best member of the population, the closeness of the second person's fitness value to the value of the Teacher is desirable.

On the other hand, this candidate's position should be far from the Teacher's position to improve the algorithm's exploration feature and prevent it from being trapped in the optimal local area. Also, the presence of this candidate in the neighborhood of the Teacher's position is undesirable because the Teacher attracts other learners to the region around him/her after the teaching phase. Consequently, the presence of this member in this region does not have a significant effect. Therefore, to select the candidate, we should trade-off between the difference of the candidate and teacher fitness value and the difference of the candidate and teacher position. A general ranking is defined as the sum of the fitness rank and the position rank as follows.

$$Rank_G = Rank_F + Rank_P, \tag{5}$$

where $Rank_F$ is fitness value rank and $Rank_P$ is position rank. The member with a lower rank value is the desirable candidate.

The range of the fitness value and position are divided into six ranks. The value of the alternative candidate should be close to the teacher value, so the closer values to the teacher value have a lower rank (lower $Rank_F$). The length of each fitness value rank (σ_F) is defined as

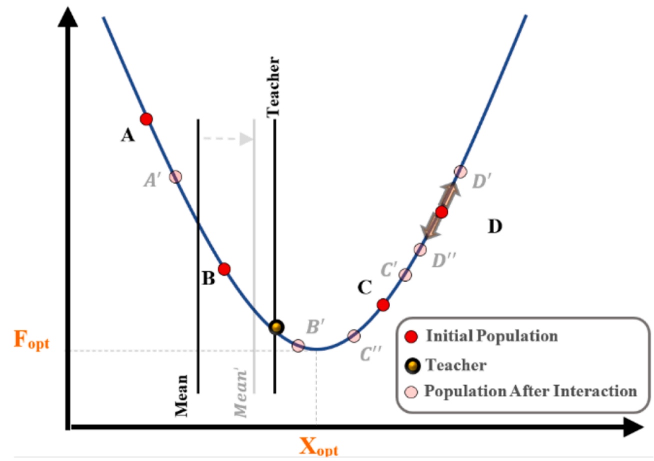

Fig. 3. Teaching phase shift strategy.

follows.

$$\sigma_F = \frac{F_{max} - F_{min}}{6}, \tag{6}$$

where F_{min} is the teacher fitness value, and F_{max} is the maximum fitness value of the population. These are the first and last values of the fitness value vector after sorting in step2. The six rank intervals for a fitness value are defined in Table 1.

A similar process is done for the position ranking. In this ranking, farther learners from the Teacher's position are ranked lower. The length of each position rank (σ_x) is defined as follows.

$$\sigma_x = \frac{max(norm(X_i - X_{Teacher}))}{6}, \tag{7}$$

where X is a position vector of each learner. For instance, in this ranking approach, the $Rank_G$ of the Teacher and its neighbors equals 7 ($Rank_P$ = 6 and $Rank_F$ = 1). Therefore, the members around the Teacher are not suitable second-member candidates. If several candidates are obtained for the second member, a candidate with the lower $Rank_P$ will be selected.

After selecting the second person, the population members are divided into two groups. One group interacts with the Teacher, and the other interacts with the second person as an alternative candidate. More members have to interact with the Teacher as the best person in the population. These numbers are defined as follows.

$$n_{2nd_{person}} = \left\lfloor \frac{F_{Teacher}}{F_{2nd_{person}}} * \frac{n}{2} \right\rfloor$$
$$n_{Teacher} = n - n_{2nd_{person}} \tag{8}$$

where $F_{2nd_{person}}$ and $F_{Teacher}$ are fitness values of second person and Teacher, respectively. If the second person's fitness value is very close to the Teacher's, half of the population should interact with the second person. Fig. 2 shows the performance of the second person as an alternative candidate. The step3 diagram in Fig. 5 shows that the modifications of this step have improved the results of the previous steps. The effectiveness of these corrections in multimodal problems is even more than unimodal.

Step4: According to Eq. (1), all members shift together in a positive or negative direction of the X-axis in the original method's teaching phase. The difference between the Teacher's position and the mean

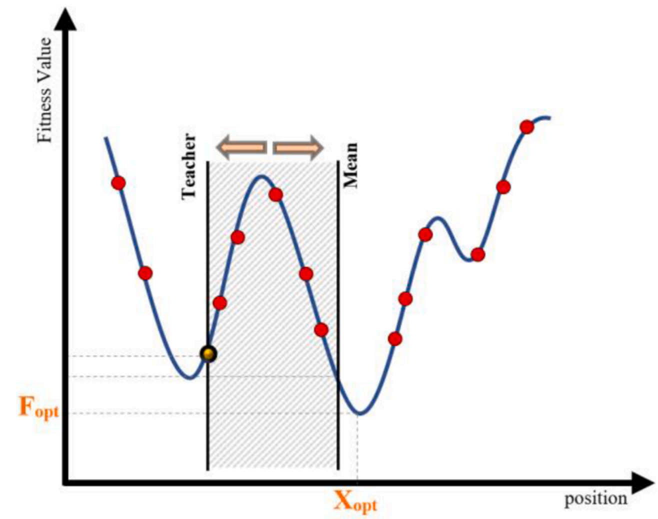

Fig. 4. Advanced teaching phase strategy.

value position determines the movement direction. The direction is reversed only in cases where $X_{teacher} > \mu$, $TF = 2$, and $TF * \mu > X_{teacher}$. In this phase, with this shift, the position of the mean value tends to be the teacher position (best answer position). As shown in Fig. 3, although the mean value of the members moves towards the best member of the population and thus towards the optimal answer with this shift (positive shift), it is observed that some members move away from the optimal answer. For instance, the C and D members move away from the optimal point $(C^{'}, D^{'})$.

This step defines an Absorbing Factor (AF) to solve this problem. This factor shifts all learners to the Teacher's position with a Δ-sized $\Delta = X_{teacher} - TF$. The right learners of the Teacher's position are shifted to the left and the left members to the right. The teaching phase formulation is modified as follows.

$$\begin{aligned} X_i^{new} &= X_i^{old} + AF|X_{teacher} - \mu| \\ AF &= sign\left(X_{teacher} - X_i^{old}\right) \end{aligned} \tag{9}$$

The teaching phase's basis is shifting the mean value position to the teacher position. If the mean fitness value is less than the teacher value in multimodal optimization problems, shifting the mean value to the teacher area in the teaching phase causes the learners to move away from the global optimal answer. As shown in Fig. 4, shifting the members between the mean value and the Teacher to the Teacher's position area is caused by deviating these members toward the local optimal. In this case, the probability of being trapped in the local optimal solutions can be reduced by guiding these learners toward the mean value's position instead of the Teacher's position. Therefore, the following conditional statement is added to the teaching phase formulation.

$$\begin{aligned} X_i^{new} &= X_i^{old} + AF|X_{teacher} - \mu| \\ AF &= \begin{cases} sign\left(X_{teacher} - X_i^{old}\right) \ if \ F(X_{teacher})\langle F(\mu) \\ sign\left(\mu - X_i^{old}\right) \ if \ F(X_{teacher})\rangle F(\mu) \end{cases} \end{aligned} \tag{10}$$

where F is the fitness function. As shown in Eq. (11), the absorbing factor of AF shifts population members to the mean value when $F(X_{teacher}) > F(\mu)$.

The Pseudo-code of the first modification part of the advanced version of TLBO is presented as follows. This pseudo-code includes four step modifications presented in the previous sections.

(continued on next column)

(continued)

Begin of Ad-TLBO Algorithm's Mechanism

Notation:
The colors of each modification of steps 1 to 4 are shown below.
Step1, Step2 , Step3 , Step4 .
npop = Initial population numbers; ***n_teacher*** = number of the population that interacts with the Teacher; ***n_2ndp*** = number of the population that interacts with the second member; ***Mean_T*** = mean value of ***n_teacher*** population; ***Mean_2ndp*** = mean value of ***n_2ndp*** population;

1:	*Generate an initial random population*
2:	*Calculate the mean value of the population*
3:	*Select Teacher*
4:	*Sort_pop=sort (pop. Cost)*
5:	$Sigma\_F = sort\_pop(end) - sort\_pop(1)$
6:	*Sort_dist=sort (norm(pop.Position − Teacher.Position))*
7:	$Sigma\_x = Sort\_dist(end)/6$
8:	***For*** *ii=1:6*
9:	*Bound_F(ii, :)=Teacher.Cost+[(ii-1) Sigma_F, (ii) Sigma_F]*
10:	*Bound_X(ii, :)=[(ii-1) Sigma_x, (ii) Sigma_x]*
11:	*RankF_loc(ii)=Find (Bound_F (ii, 1)≤ pop. Cost≤ Bound_F(ii,2)*
12:	*RankP_loc(ii)=Find (Bound_X (ii, 1)≤ Sort_dist ≤ Bound_X(ii,2)*
13:	*Rank_F (RankF_loc(ii)) =ii; Rank_P (RankP_loc(ii)) =6-(ii-1);*
14:	***END***
15:	*[RankG.Cost, RankG.position] =sort (Rank_F+Rank_P)*
16:	*Second_person.Cost=pop (RankG.position).Cost*
17:	*Second_person.Position=pop (RankG.position).Position*
18:	*If size (Second_person.Cost) ≥ 2 select the lower rank person*
19:	*n_2ndp=ceil(Teacher.Cost/Second_person.Cost ∗ npop/2)*
20:	*n_teacher = npop − n_2ndp*
21:	*pop_2ndp= Select n_2ndp random number of the population.*
22:	*pop_T= n_teacher remaining members of the population.*
23	*Calculate mean values of n_2ndp and n_teacher*
	Begin of Teaching Phase
24	***For*** *i=1: n_teacher*
25	*AF_T=sign(Teacher.Position-pop_T(i).Position)*
26	***If*** *Cost(Mean_T)< Cost(Teacher.Cost)*
27	*AF_T=sign(Mean_T-pop_T(i).Position)*
28	***End***
29	*Newsol(i).Position_T= pop_T(i).Position+AF_T*abs(Teacher.Position-Mean_T)*
30	***End***
31	***For*** *i=1: n_2ndp*
32	*AF_2ndp=sign(_second_person.Position-pop_2ndp(i).Position)*
33	***If*** *Cost(Mean_2ndp)< Cost(second_person.Cost)*
34	*AF_2ndp=sign(Mean_2ndp -pop_2ndp(i).Position)*
35	***End***
36	*Newsol(i).Position_2ndp= pop_2ndp(i).Position+AF_2ndp*abs(Teacher.Position-Mean_2ndp)*
37	***End***
38	***Gathering and sorting pre-and post-interaction members***
39	*All_pop=[pop, Newsol_T, Newsol_2ndp]*
40	*All_pop=sort(All_pop);*
41	*Pop= All_pop(1:npop) (select first npop of All_pop members for new population)*
	Begin of Learning Phase
42	*Select a new Teacher for the new population after the teaching phase*
43	*Repeat lines 4 to 22 again to find the second member*
44	***For*** *i=1: n_teacher*
45	*Shift_stepT= Teacher.Position-pop_T(i).Position*
46	*Newsol(i).Position_T= pop_T(i).Position+r*TF(Shift_stepT)*
47	***End***
48	***For*** *i=1: n_2ndp*
49	*Shift_step2ndp= Second_person.Position-pop_2ndp(i).Position*
50	*Newsol(i).Position_2ndp= pop_2ndp(i).Position+r*TF(Shift_step2ndp)*
51	***End***
52	*Repeat lines 38 to 41 again for selecting a new population from members before and after the teaching phase.*
53	*Check termination criteria.*
54	*If that condition is satisfied, the algorithm terminates; otherwise, the algorithm repeats with the new population.*

Fig. 5 shows that the proposed Ad-TLBO significantly improves the O-TLBO's performance. This advancement is obtained from the modifications collection of the previous four steps; therefore, the synergy of these advantages leads to reaching a much better solution than the original method.

Part II

In the second part, the generation of the initial population and the exploration approach of the O- TLBO is improved. This part is not limited to the TLBO method and applies to other methods. As shown in Fig. 5 and other subsequent challenges, the Ad-TLBO method has a high convergence rate in low iteration, so the changes in this part are more effective in this method. The basis of these changes is the segmentation of the search space. The first change is how to use the algorithm to search the space and the second one is a kind of intelligent generation of the initial population.

Classified search

This approach mimics the strategy of a search team to find the deepest valley in a vast land in a limited time. In this strategy, team

Beginning the Classified Search's Mechanism	
Notation: The colors of each section are shown below. Section1 ,Section2 , Section3 .	
1:	*Calculate MEF_n of each section by Eq 12*
2:	*Calculate MEF_n of each hypercube of each section by Eq 13*
3:	*Determine the number of populations of each hypercube of each section (npop1, npop2, npop3)*
4:	*Determine the number of "nc"*
	Beginning Section-1
5:	*Generation of row vectors with nvar elements from all possible permutations consisting of numbers 1 or 2 by a permutations function*
6:	***For** i=1:2^nvar*
7:	*Making the boundaries of i-th hypercube according to the i-th row of output matrix of permutations function in line 5.*
8:	*Generation of a random initial population with npop1 members within the boundaries of the i-th hypercube*
9:	*Run ATLBO algorithm for i-th hypercube of search space with $MEF_n hc1$ function evaluation*
10:	***END***
11:	*Sorting the best answers of optimization output for each hypercube and selecting the "nc" number of the best hypercubes for the second section.*
	Beginning Section-2
12:	*Ratio=sort_cost(1:nc)/sort_cost (1)*
13:	***For** j=1:nc*
14:	***If** Ratio(j)>=2*
15:	*nc=j*
16:	***END** (if)*
17:	***END** (for)*
18:	***For** k=1:nc*
19:	*Generation of a random initial population with npop2 members within the boundaries of the k-th hypercube. (For the first hypercube (npop2-1) members is generated.)*
20:	*Run ATLBO algorithm for k-th hypercube of search space with $MEF_n hc2$ function evaluation*
21:	***END***
22:	*Sorting the best answers of optimization output for each hypercube and selecting the best for the third section.*
	Beginning the third section
23	*Generation of a random initial population with (npop3-1) members.the best member of section 2 is added to the initial population of section 3.*
24	*Run ATLBO algorithm with $MEF_n hc3$ function evaluation*

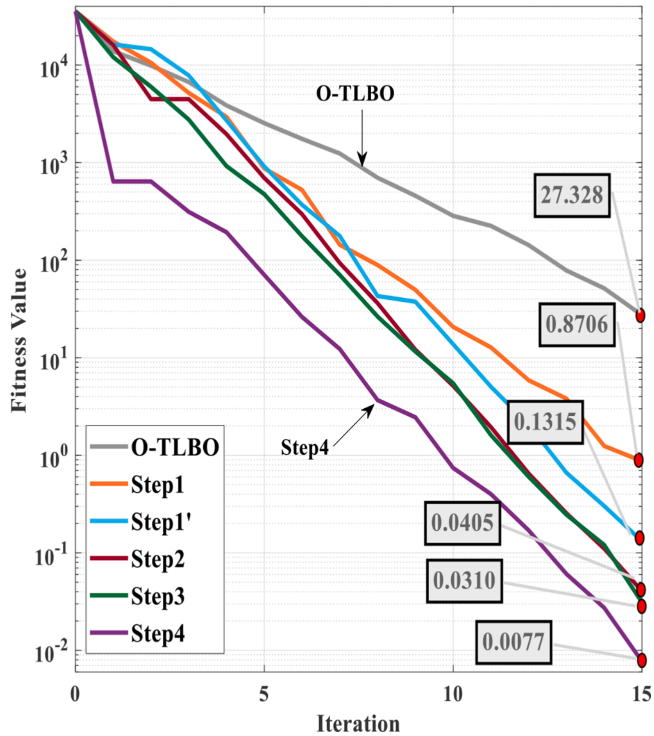

Fig. 5. Sphere benchmark best fitness value by O-TLBO and Ad-TLBO.

members are divided into several smaller groups and search in different parts of the land for a fraction of the available time. All team members will search these few candidate areas in the second part of the available time. After comparing the results, the search area is reduced to a few smaller parts. All team members will search these few candidate areas in the second part of the available time. Finally, in the last section of the remaining available time, all members focus on one candidate part with the best result.

In the proposed approach the Maximum Evaluation Function number (MEF_n) is similarly divided into three sections instead of available time. Each variable is divided into two equal parts in the first section, so the search space is divided into 2^{nvar} smaller hypercubes. K_1% of $MEF_n$is allocated to this section of the search. In the second section, K_2% of $MEF_n$is allocated to searching in the "nc" number of candidate hypercubes with better results. Finally, the remaining $MEF_n$is allocated to searching for the best hypercube in the last section. The allocated $MEF_n$of each section is calculated as follows.

$$\begin{aligned} MEF_n1 &= K_1 MEF_n \\ MEF_n2 &= K_2 MEF_n \\ MEF_n3 &= (1-(K_2+K_1))MEF_n \end{aligned} \tag{11}$$

In this approach, searching the entire solution space through segmentation improves the exploration of the algorithm. The algorithm's exploitation also improves as the search space is reduced to $1\ /2^{nvar}$ in the last part. For instance, the search space of the 3-variable problem is divided into eight smaller cubes (Fig. 6). The procedure of the other two sections is also shown in Fig. 6.

The allocated $MEF_n$of each hypercube of each section is calculated as follows.

$$\begin{aligned} MEF_nhc1 &= [MEF_n1/2^{nvar}] \\ MEF_nhc2 &= [MEF_n2/nc] \\ MEF_nhc3 &= MEF_n3 \end{aligned} \tag{12}$$

The Pseudo-code of this Classified Search is presented as follows.

The output of the permutation function in line 5 of the Pseudo-code is a 2^{nvar}-by-$nvar$ matrix. Each matrix row represents a hypercube location coordinating in the search space. As mentioned, the range of each variable is divided into two parts. The number 1 indicates the selection from the first half of the range, and the number 2 indicates the selection from the second half. Lines 12 to 17 of the Pseudo-code add a filter to the process. This filter eliminates hypercubes that are very larger than the best answer from Section 2 candidates. Therefore, the additional $MEF_n$is eliminated by decreasing the number of "nc". In line 19, for the first hypercube (best hypercube of section1), ($npop2-1$) members are generated. In this line, the best member of the first hypercube (the best hypercube) from section 1 is added to the members of the initial population in section 2. The main application of this approach is in complicated problems such as highly multimodal, non-separable, and multivariable problems. This type of

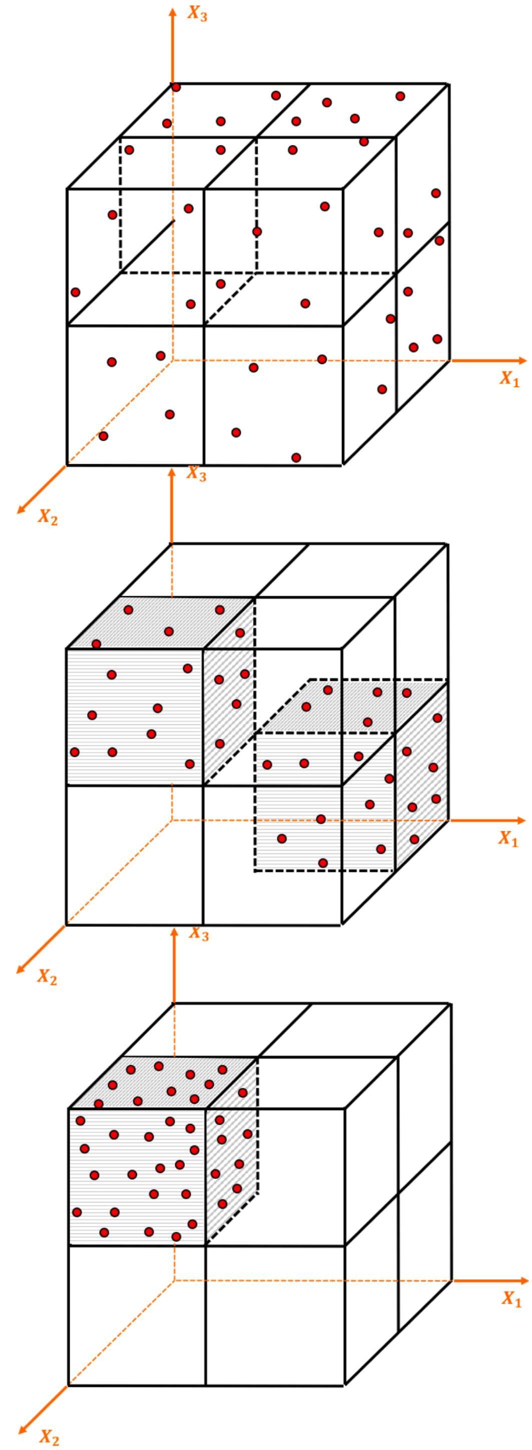

Fig. 6. Classified Search strategy.

search can be inactivated for simple problems

Generation with whole search space coverage

Instead of randomly generating the initial population, this section proposes an intelligent generation approach to cover the entire search space. This approach divides the search space into H equal hypercubes and puts at least one member in each hypercube. This initialization technique reduces the probability of being trapped in local solutions by covering the search space. For the *nvar*-variable problem with the *npop* number initial population, the value of H is calculated as follows.

$$h = \lfloor \sqrt[nvar]{npop} \rfloor = \lfloor npop^{1/nvar} \rfloor \qquad H = h^{nvar} \tag{13}$$

where h is the number of divisions in each dimension. For example, in the three-variable problem with a population number of 100, *h* equals 4. As shown in Fig. 7, cube search space is made up of 64 (4^3) smaller cubes. According to the proposed approach, one value must be placed in each cube. Therefore, 64 out of 100 members are placed into these small cubes in this example. The remaining 36 numbers can be placed using a uniform distribution in the search space. The range of each section Δ is defined as follows.

Beginning the Search space Coverage's Mechanism

Notation:
The colors of each section are shown below.
Generation of the members of each cube (green). Additional members generation (orange).
pop0= *Custom members of the initial population,* ***npop0=*** Number of *custom members of the initial population.* ***lb****= Vector of lower bound of search space.* ***ub****= Vector of the upper bound of search space.* ***Data0****= matrix of all hypercubes members that are classified as wall-to-wall.*

```
Input npop, nvar,pop0,lb,ub
Calculate h, H by Eq 13
For j=1:h
    Calculate the boundaries of the hypercubes of each
    variable by Eq15
    Data0= Generate the population members wall-to-wall
           (described in the text) in the boundaries of
           each hypercube by the uniform distribution
           and then store them in a matrix.
END
Loc=Generate all permutations numbers from 1 to h for
    nvar variables
For k1=1:H
  For k2=1:h
    For k3=1:nvar
      Place the member from Data0 corresponding to
      the coordinates of each cube (loc) in the initial
      population
    END (for k3)
  END (for k2)
END (for k1)
Additional_num= npop-H, npop0= size(pop0,1)
Additional_pop=pop0
For i=1: Additional_num-npop0
    Additional_pop(npop0+i,:)=unifrnd(lb,ub,[1,nvar])
END
Initial_Population= [Initial_Population, Additional_pop]
```

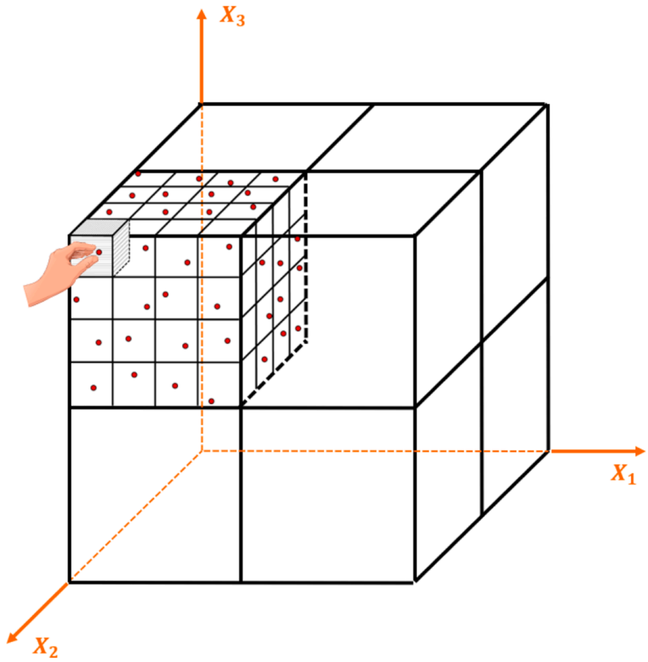

Fig. 7. Initial population generation with the better search space coverage.

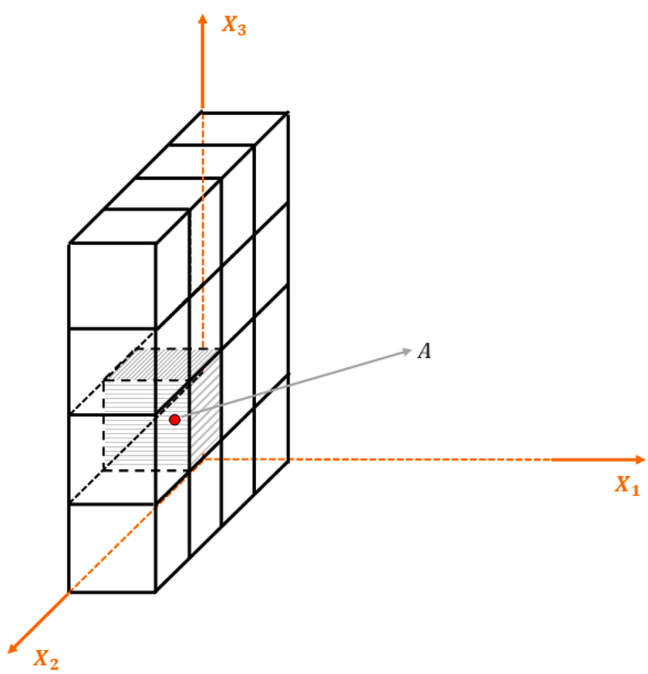

Fig. 8. A constant interval of search space.

$$\Delta = \frac{U_b - L_b}{h}, \tag{14}$$

where U_b and L_b are the vectors of upper and lower bounds of the variables, respectively. A uniform distribution randomly generates each member in each cube to maintain the random nature of the initial population generation. The boundaries of the hypercubes for each variable are calculated as follows.

Table 2
Differences between Ad-TLBO with O-TLBO.

Concepts	O-TLBO	Ad-TLBO
Initialization Technique	Generation of a simple randomness initial population in the range of [Lb, Ub]	Whole search space coverage approach
Search Technique	Iterative steps	Classified Search
Teaching Phase	Eq. (1)	**Step2:** Eliminate binary comparison and keep valuable members with a better approach **Step3:** Select a second member (alternative Teacher) by ranking approach **Step4:** Modifying the movement direction of the learners by the Absorbing Factor in Eq. (9) and adding the conditional statement in Eq. (10)
Learning Phase	Eq. (2)	**Step1:** 1. Interact only with the Teacher instead of random interactions or interactions with all learners. 2. selecting a new teacher after the teaching phase (Eq. (4)) **Step2:** Eliminate binary comparison and keep valuable members with a better approach **Step3:** Select a second member (alternative Teacher) by ranking approach

$$[\,Lb_c \quad Ub_c\,] = \begin{bmatrix} L_b + [0, \Delta] \\ L_b + [\Delta, 2\Delta] \\ \vdots \\ L_b + [(k-1)\Delta, U_b] \end{bmatrix}, \tag{15}$$

For example, in a three-dimensional problem where $h = 4$, the presented wall in Fig. 8 is all the hypercubes in the constant interval of $[L_{b1}, L_{b1} + \Delta_1]$ for the variable of X_1. As indicated in this figure, 16 small cubes make this wall, and in every 16 cubes, X_1 is in the interval $[L_{b1}, L_{b1} + \Delta_1]$. Hence, 16 uniform random numbers must be generated in this interval. In general, for the nvar-dimensional problem, $h^{(nvar-1)}$ uniform random numbers must be generated in this interval. For this example, the intervals of all walls for the variable X_1 are as follows.

$$[L_{b1}, L_{b1} + \Delta_1], [L_{b1} + \Delta_1, L_{b1} + 2\Delta_1], \ [L_{b12} + 2\Delta_1, L_{b1} + 3\Delta_1], [L_{b1} + 3\Delta_1, U_{b1}] \tag{16}$$

Similarly, the intervals of each wall of the variables X_2, X_3 must be determined.

According to lines 7 to 10 of the Pseudo-code below, this approach can be implemented using a permutation function and assigning a coordinate to each hypercube. To generate member A in Fig. 8, according to the coordinates of the shaded hypercube, three random numbers must be generated in three intervals by the uniform distribution as follows.

Table 3
Best fitness value in each step.

Method	Best Value
O-TLBO	**27.328**
Step1	**0.8706**
Step1′	**0.1315**
Step2	**0.0405**
Step3	**0.0310**
Ad-TLBO	**0.0077**

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}_A = Rand\left(\begin{bmatrix} [L_{b1}, L_{b1} + \Delta_1] \\ [L_{b2} + 2\Delta_2, L_{b2} + 3\Delta_2] \\ [L_{b3} + \Delta_3, L_{b3} + 2\Delta_3] \end{bmatrix}\right) \tag{17}$$

Note that no additional evaluation function is added to the algorithm in the proposed approach. In the original version, pop random numbers with the bounds of a large cube (boundaries of variables) made the initial population. However, each random number is made inside a smaller cube with this change. The proposed approach pseudo-code is shown below.

Table 2 shows the summarized difference between Ad-TLBO with the original version.

Challenges

In what follows, four challenges are used to investigate the performance of the proposed advanced algorithm (Ad-TLBO). In each of these challenges, some known benchmarks in the literature are utilized for different purposes. The proposed algorithm is compared with the original version and some of the most widely used intelligent optimization algorithms with different parameters in these benchmarks. The proposed Classified Search approach is only used in the last two challenges and is inactive in other challenges.

Challenge1

In the first challenge, a sphere benchmark shows the effect of modifying each step mentioned in the previous section compared to the original version or the steps before. The sphere benchmark defined in (18) is a minimization problem with the global optimum solution equal to zero at $= [0, 0\ldots,\ 0]$:

$$f(x_i) = \sum_{i=1}^{n} x_i^2 - 100 \le x_i \le 100 \tag{18}$$

Table 4
Benchmarks characteristics.

	Interval	Global optimum	Multimodal	Separable	Regular
Sphere	**[−100,100]**	$F = 0$, $X = (\vec{0})$	**No**	**Yes**	**Yes**
Rastrigin	**[−5.12,5.12]**	$F = 0$, $X = (\vec{0})$	**Yes**	**Yes**	**Yes**
Griewank	**[−600,600]**	$F = 0$, $X = (\vec{0})$	**Yes**	**No**	**Yes**
Rosenbrock	**[−2.048,2.048]**	$F = 0$, $X = (\vec{1})$	**No**	**No**	**Yes**
Zakharov	**[−10,10]**	$F = 0$, $X = (\vec{0})$	**No**	**Yes**	**Yes**
Ackley	**[−32.76,32.76]**	$F = 0$, $X = (\vec{0})$	**Yes**	**No**	**Yes**
Weierstrass	**[−0.5,0.5]**	$F = 0$, $X = (\vec{0})$	**Yes**	**Yes**	**No**
Quadric	**[−100,100]**	$F = 0$, $X = (\vec{0})$	**Yes**	**Yes**	**Yes**
Sum Square	**[−100,100]**	$F = 0$, $X = (\vec{0})$	**No**	**Yes**	**Yes**

Table 5
Accuracy comparison of various algorithms for 10-dimensional functions. Except for Ad-TLBO and A-TLBO-DA, the rest of the results are taken from Ref. Chen et al. (2016a).

		Sphere	Quadric	Sum Square	Zakharov	Rosenbrock	Ackley	Rastrigin	Weierstrass	Griewank
DE	**Mean**	**7.13e-073**	**4.18e-012**	**2.47e-074**	**1.08e-005**	**6.24E+000**	**3.48e-015**	**1.46e+000**	**0.00e+000**	**3.50e-002**
	STD	**7.18e-073**	**9.35e-012**	**4.86e-074**	**2.41e-005**	**9.02e-001**	**2.37e-016**	**5.02e-001**	**0.00e+000**	**2.22e-002**
jDE	**Mean**	**1.31e-076**	**1.14e-021**	**6.97e-078**	**1.31e-031**	**5.14e-007**	**3.36e-015**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	STD	**1.58e-076**	**1.52e-021**	**1.39e-077**	**1.30e-031**	**9.47e-007**	**4.27e-016**	**0.00e+000**	**0.00e+000**	**0.00e+000**
SaDE	**Mean**	**1.35e-071**	**1.89e-019**	**1.28e-074**	**6.65e-031**	**2.62e+000**	**3.28e-015**	**0.00e+000**	**0.00e+000**	**1.48e-003**
	STD	**2.02e-071**	**3.54e-019**	**2.52e-074**	**1.48e-030**	**1.50e+000**	**2.51e-016**	**0.00e+000**	**0.00e+000**	**3.31e-003**
PSOwFIPS	**Mean**	**3.98e-016**	**6.19e-006**	**2.18e-017**	**3.23e-009**	**4.51e+000**	**8.04e-009**	**1.89e+000**	**4.24e-004**	**7.59e-002**
	STD	**6.09e-016**	**2.15e-006**	**1.39e-017**	**2.23e-009**	**7.17e-002**	**4.33e-009**	**1.03e+000**	**6.60e-004**	**5.19e-002**
CLPSO	**Mean**	**1.09e-018**	**5.37e-001**	**2.59e-020**	**2.66e-003**	**2.45e+000**	**4.28e-010**	**2.76e-009**	**6.33e-012**	**4.15e-003**
	STD	**1.5e-018**	**1.38e-001**	**1.76e-020**	**2.37e-003**	**1.00e+000**	**2.89e-010**	**3.94e-009**	**6.08e-012**	**5.68e-003**
ABC	**Mean**	**8.02e-017**	**4.04e+001**	**7.18e-017**	**1.31e+001**	**2.84e-001**	**8.53e-015**	**0.00e+000**	**0.00e+000**	**3.94e-003**
	STD	**3.22e-017**	**2.35e+001**	**3.94e-017**	**8.38e+000**	**3.23e-001**	**3.18e-015**	**0.00e+000**	**0.00e+000**	**5.67e-003**
TLBO	**Mean**	**3.29e-184**	**2.56e-082**	**9.94e-187**	**1.51e-089**	**4.96e-001**	**3.43e-015**	**3.06e+000**	**0.00e+000**	**6.48e-003**
	STD	**3.08e-185**	**5.58e-082**	**1.25e-187**	**1.62e-089**	**4.21e-001**	**217e-015**	**1.52e+000**	**0.00e+000**	**9.71e-003**
ATLBO-DA	**Mean**	**4.84e-013**	**0.00e+000**	**1.82e-012**	**7.31e-013**	**2.11e-003**	**7.04e-007**	**2.34e+000**	**5.70e-014**	**5.86e-003**
	STD	**4.50e-013**	**0.00e+000**	**1.88e-012**	**7.12e-013**	**1.15e-002**	**3.05e-007**	**5.74e+000**	**1.30e-014**	**1.81e-002**
ETLBO	**Mean**	**2.84e-166**	**3.22e-079**	**6.50e-169**	**2.94e-087**	**1.46e-001**	**3.37e-015**	**3.02e+000**	**0.00e+000**	**2.42e-002**
	STD	**4.27e-167**	**5.07e-079**	**5.49e-170**	**3.10e-087**	**1.38e-001**	**1.05e-015**	**1.86e+000**	**0.00e+000**	**3.69e-002**
sawTLBO	**Mean**	**3.01e-064**	**4.59e-050**	**4.86e-067**	**6.87e-053**	**1.90e+000**	**2.84e-015**	**8.57e+000**	**0.00e+000**	**1.33e-002**
	STD	**4.86e-064**	**1.12e-049**	**1.54e-066**	**1.61e-052**	**5.28e-001**	**1.50e-015**	**3.23e+000**	**0.00e+000**	**2.22e-002**
VTLBO	**Mean**	**3.56e-296**	**3.50e-130**	**4.63e-298**	**1.02e-139**	**1.13e+000**	**1.78e-015**	**1.09e+000**	**0.00e+000**	**3.82e-008**
	STD	**0.00e+000**	**5.05e-130**	**0.00e+000**	**3.21e-139**	**5.06e-001**	**1.87e-015**	**1.52e+000**	**0.00e+000**	**1.21e-007**
Ad-TLBO	**Mean**	**0.00e+000**	**5.66e-221**	**0.00e+000**	**0.00e+000**	**3.53e-032**	**1.30e-016**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	STD	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**	**1.82e-031**	**7.15e-016**	**0.00e+000**	**0.00e+000**	**0.00e+000**

Table 6
Accuracy comparison of various algorithms for 30-dimensional functions. Except for Ad-TLBO and A-TLBO-DA, the rest of the results are taken from Ref. Chen et al. (2016a).

		Sphere	Quadric	Sum Square	Zakharov	Rosenbrock	Ackley	Rastrigin	Weierstrass	Griewank
DE	**Mean**	**4.90e-014**	**4.14e+000**	**7.99e-017**	**8.79e-001**	**2.54e+001**	**7.49e-009**	**7.75e+001**	**1.20e-002**	**3.94e-003**
	STD	**1.08e-013**	**3.22e+000**	**8.83e-017**	**6.15e-001**	**5.86e-001**	**5.58e-009**	**3.01e+001**	**2.67e-002**	**5.40e-003**
jDE	**Mean**	**1.95e-022**	**2.06e+001**	**3.92e-023**	**1.35e+000**	**2.18e+001**	**2.82e-012**	**2.24e-009**	**3.16e-001**	**0.00e+000**
	STD	**2.76e-022**	**6.71e+000**	**3.86e-023**	**1.68e+000**	**2.59e-001**	**1.77e-012**	**4.15e-009**	**5.07e-001**	**0.00e+000**
SaDE	**Mean**	**3.84e-023**	**1.06e+001**	**3.00e-024**	**1.50e-001**	**2.52e+001**	**1.30e-012**	**5.53e-001**	**6.56e-011**	**0.00e+000**
	STD	**2.15e-023**	**6.53e+000**	**2.47e-024**	**1.31e-001**	**1.36e+000**	**8.26e-013**	**7.55e-001**	**8.87e-011**	**0.00e+000**
PSOwFIPS	**Mean**	**1.43e+000**	**3.82e+003**	**2.17e-001**	**1.11e+002**	**2.73e+001**	**5.12e-001**	**1.23e+002**	**2.46e+000**	**8.92e-001**
	STD	**2.78e-001**	**1.01e+003**	**7.69e-002**	**1.89e+001**	**2.99e-001**	**1.78e-001**	**1.57e+001**	**7.11e-001**	**5.53e-002**
CLPSO	**Mean**	**1.94e-001**	**1.16e+004**	**2.25e-002**	**2.30e+002**	**6.84e+001**	**7.13e-001**	**2.91e+001**	**5.18e-001**	**2.68e-001**
	STD	**7.79e-002**	**2.72e+003**	**6.43e-003**	**5.45e+001**	**2.84e+001**	**5.74e-001**	**4.77e+000**	**6.83e-002**	**4.38e-002**
ABC	**Mean**	**2.45e-006**	**1.52e+004**	**6.88e-007**	**5.48e+002**	**2.16e+001**	**9.08e-003**	**4.22e+000**	**3.66e-002**	**3.32e-003**
	STD	**1.21e-006**	**2.77e+003**	**3.24e-007**	**6.58e+001**	**4.59e+000**	**4.76e-003**	**5.91e-001**	**8.53e-003**	**6.99e-003**
TLBO	**Mean**	**4.04e-111**	**1.08e-022**	**5.38e-111**	**5.11e-011**	**2.38e+001**	**3.55e-015**	**1.17e+001**	**0.00e+000**	**0.00e+000**
	STD	**3.20e-111**	**1.43e-022**	**3.43e-111**	**8.59e-011**	**7.01e-001**	**0.00e+000**	**3.71e+000**	**0.00e+000**	**0.00e+000**
ATLBO-DA	**Mean**	**1.18e-012**	**0.00e+000**	**1.17e-011**	**1.78e-012**	**4.16e-004**	**6.40e-007**	**2.78e-010**	**1.82e-013**	**4.07e-014**
	STD	**4.50e-013**	**0.00e+000**	**1.11e-011**	**1.52e-012**	**2.28e-003**	**3.29e-007**	**1.97e-010**	**4.20e-014**	**3.14e-014**
ETLBO	**Mean**	**2.66e-095**	**3.42e-022**	**8.21e-096**	**1.92e-011**	**2.38e+001**	**3.55e-015**	**1.22e+001**	**0.00e+000**	**0.00e+000**
	STD	**1.84e-095**	**4.72e-022**	**1.11e-095**	**1.93e-011**	**8.57e-001**	**0.00e+000**	**9.07e+000**	**0.00e+000**	**0.00e+000**
sawTLBO	**Mean**	**2.61e-065**	**1.68e-026**	**6.13e-066**	**1.23e-008**	**2.36e+001**	**3.55e-015**	**2.63e+001**	**0.00e+000**	**0.00e+000**
	STD	**3.09e-065**	**1.78e-026**	**1.12e-065**	**2.54e-008**	**8.68e-001**	**0.00e+000**	**6.04e+000**	**0.00e+000**	**0.00e+000**
VTLBO	**Mean**	**4.85e-158**	**1.24e-032**	**2.50e-158**	**8.06e-018**	**2.28e+001**	**3.55e-015**	**1.25e+001**	**0.00e+000**	**0.00e+000**
	STD	**1.06e-157**	**1.79e-032**	**2.48e-158**	**1.11e-017**	**4.23e-001**	**0.00e+000**	**7.27e+000**	**0.00e+000**	**0.00e+000**
Ad-TLBO	**Mean**	**0.00e+000**	**3.69e-033**	**0.00e+000**	**1.15e-048**	**6.98e-033**	**3.02e-015**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	STD	**0.00e+000**	**2.02e-032**	**0.00e+000**	**4.28e-048**	**2.32e-032**	**1.08e-015**	**0.00e+000**	**0.00e+000**	**0.00e+000**

Table 7
Convergence rate and reliability comparison of various algorithms for 10-dimensional functions. Except for Ad-TLBO and A-TLBO-DA, the rest of the results are taken from Ref. Chen et al. (2016a).

	Acceptance Value	Sphere 1e-6	Quadric 1e-6	Sum Square 1e-6	Zakharov 1e-6	Rosenbrock 0.1	Ackley 1e-6	Rastrigin 2.5	Weierstrass 1e-6	Griewank 0.1
DE	**mFEs**	**6733**	**11,892**	**5947**	**9372**	**NaN**	**10,121**	**31,720**	**15,015**	**11,865**
	success	**100%**	**100%**	**100%**	**100%**	**0%**	**100%**	**100%**	**100%**	**100%**
jDE	**mFEs**	**6218**	**17,719**	**5628**	**13,083**	**30,868**	**9619**	**6297**	**16,636**	**5231**
	success	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
SaDE	**mFEs**	**6395**	**19,236**	**5635**	**13,163**	**NaN**	**9586**	**9132**	**12,775**	**8896**
	success	**100%**	**100%**	**100%**	**100%**	**0%**	**100%**	**100%**	**100%**	**100%**
PSOwFIPS	**mFEs**	**24,291**	**NaN**	**20,803**	**37,939**	**NaN**	**38,944**	**39,383**	**NaN**	**32,582**
	success	**100%**	**0%**	**100%**	**100%**	**0%**	**100%**	**80.3%**	**0%**	**80.5%**
CLPSO	**mFEs**	**27,464**	**NaN**	**24,573**	**NaN**	**NaN**	**37,753**	**28,860**	**41,247**	**23,562**
	success	**100%**	**0%**	**100%**	**0%**	**0%**	**100%**	**100%**	**100%**	**100%**
ABC	**mFEs**	**11,300**	**NaN**	**9627**	**NaN**	**46,110**	**21,143**	**8529**	**25,594**	**7392**
	success	**100%**	**0%**	**100%**	**0%**	**60.3%**	**100%**	**100%**	**100%**	**100%**
TLBO	**mFEs**	**2728**	**5659**	**2400**	**5814**	**NaN**	**4126**	**27,555**	**6093**	**5084**
	success	**100%**	**100%**	**100%**	**100%**	**0%**	**100%**	**40.7%**	**100%**	**100%**
ATLBO-DA	**mFEs**	**2482.7**	**634.7**	**2558.7**	**3424**	**1334.7**	**4940**	**4854.7**	**238.7**	**2740**
	success	**100%**	**100%**	**100%**	**100%**	**100%**	**86.7%**	**83.3%**	**100%**	**100%**
ETLBO	**mFEs**	**3038**	**5968**	**2584**	**6062**	**44,657**	**4563**	**41,698**	**6604**	**5604**
	success	**100%**	**100%**	**100%**	**100%**	**60.8%**	**100%**	**20.4%**	**100%**	**100%**
sawTLBO	**mFEs**	**2381**	**4775**	**2066**	**4863**	**NaN**	**3472**	**NaN**	**4876**	**9030**
	success	**100%**	**100%**	**100%**	**100%**	**0%**	**100%**	**0%**	**100%**	**100%**
VTLBO	**mFEs**	**1201**	**3343**	**975**	**3563**	**NaN**	**2089**	**15,596**	**3817**	**1940**
	success	**100%**	**100%**	**100%**	**100%**	**0%**	**100%**	**80.6%**	**100%**	**100%**
Ad-TLBO	**mFEs**	**539.33**	**2097**	**635.33**	**1268**	**309.3**	**2223.7**	**735**	**1531**	**981.6**
	success	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**

This challenge is only to demonstrate the performance improvement of the proposed algorithm in each step. In the following challenges, the effectiveness of the proposed algorithm can be seen in different benchmarks compared to the original version and other optimization algorithms statistically.

As seen in Fig. 5 and Table 3, at the same condition (iterations = 15, populations number = 50, and the number of variables = 20), the performance of the algorithm is improved step by step compared to the initial version. Note that the final response accuracy of the algorithm is about 3500 times better than the original version in the fourth step. Also, the slopes of the diagrams show that the convergence rate is improved step by step. It should be noted that changes are added to the previous steps modifications in each step.

Challenge2

In this challenge, the performance of the proposed algorithms in four characteristics of accuracy, convergence rate, reliability, and statistical analysis is compared to some other algorithms using nine benchmark functions.

The selected benchmarks are different in multimodality, separability, and differentiability characteristics. Variations in these characteristics can seriously challenge the ability of the algorithm to find the optimal global solution. Table 4 shows the benchmark's characteristics.

Algorithms of Differential Evolution (DE) and its variants (self-adaptive DE (SaDE) and jDE algorithm), PSO variants, and TLBO variants are investigated to compare them with the proposed algorithm.

Table 8
Convergence rate and reliability comparison of various algorithms for 30-dimensional functions. Except for Ad-TLBO and A-TLBO-DA, the rest of the results are taken from Ref. Chen et al. (2016a).

	Acceptance Value	Sphere 1e-6	Quadric 1e-6	Sum Square 1e-6	Zakharov 1e-6	Rosenbrock 0.1	Ackley 1e-6	Rastrigin 2.5	Weierstrass 1e-6	Griewank 0.1
DE	**mFEs**	**27,150**	**NaN**	**24,454**	**NaN**	**NaN**	**39,175**	**NaN**	**NaN**	**15,701**
	success	**100%**	**0%**	**100%**	**0%**	**0%**	**100%**	**0%**	**0%**	**100%**
jDE	**mFEs**	**20,132**	**NaN**	**18,788**	**NaN**	**NaN**	**29,075**	**29,888**	**NaN**	**11,780**
	success	**100%**	**0%**	**100%**	**0%**	**0%**	**100%**	**100%**	**0%**	**100%**
SaDE	**mFEs**	**19,179**	**NaN**	**17,115**	**NaN**	**NaN**	**27,726**	**45,886**	**39,975**	**11,037**
	success	**100%**	**0%**	**100%**	**0%**	**0%**	**100%**	**100%**	**100%**	**100%**
PSOwFIPS	**mFEs**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**
	success	**0%**	**0%**	**0%**	**0%**	**0%**	**0%**	**0%**	**0%**	**0%**
CLPSO	**mFEs**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**
	success	**0%**	**0%**	**0%**	**0%**	**0%**	**0%**	**0%**	**0%**	**0%**
ABC	**mFEs**	**NaN**	**NaN**	**48,083**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**29,454**
	success	**0%**	**0%**	**60%**	**0%**	**0%**	**0%**	**0%**	**0%**	**100%**
TLBO	**mFEs**	**4724**	**19,289**	**4397**	**36,703**	**NaN**	**6813**	**NaN**	**9809**	**2808**
	success	**100%**	**100%**	**100%**	**100%**	**0%**	**100%**	**0%**	**100%**	**100%**
ATLBO-DA	**mFEs**	**3066.7**	**881.3**	**3401.3**	**5465.3**	**1010.7**	**5173.3**	**2762.7**	**562.7**	**1301.3**
	success	**100%**	**100%**	**100%**	**100%**	**100%**	**90%**	**100%**	**100%**	**100%**
ETLBO	**mFEs**	**5521**	**19,131**	**51,119**	**36,607**	**NaN**	**8024**	**9069**	**11,236**	**3192**
	success	**100%**	**100%**	**100%**	**100%**	**0%**	**100%**	**20%**	**100%**	**100%**
sawTLBO	**mFEs**	**4065**	**16,458**	**3833**	**38,919**	**NaN**	**5730**	**NaN**	**7773**	**2495**
	success	**100%**	**100%**	**100%**	**100%**	**0%**	**100%**	**0%**	**100%**	**100%**
VTLBO	**mFEs**	**2896**	**13,656**	**2571**	**26,751**	**NaN**	**4580**	**3841**	**7395**	**1663**
	success	**100%**	**100%**	**100%**	**100%**	**0%**	**100%**	**20.5%**	**100%**	**100%**
Ad-TLBO	**mFEs**	**1116.7**	**2135**	**1222**	**8476**	**330.6**	**3843**	**2676.3**	**2977.3**	**1210**
	success	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**

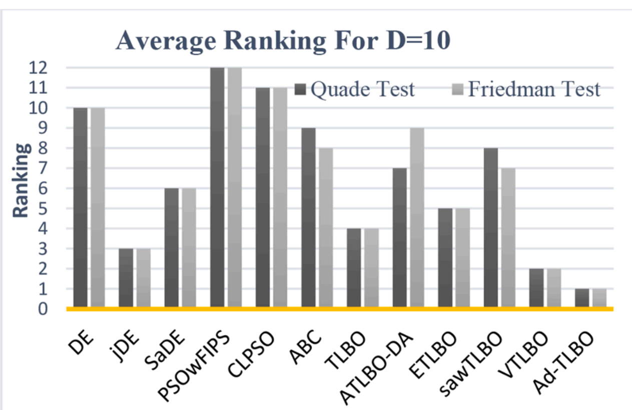

Fig. 9. Average ranking in Quade and Friedman tests for $D = 10$.

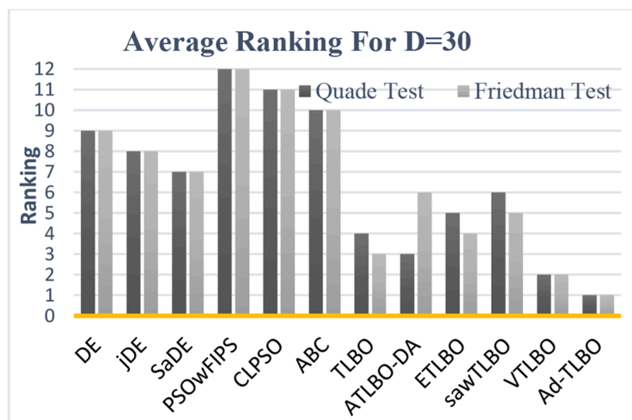

Fig. 10. Average ranking in Quade and Friedman tests for $D = 30$.

Accuracy: The accuracy characteristics of the investigated algorithms in terms of mean and standard deviation (STD) are shown in Tables 5 and 6 for 10- and 30-dimension problems, respectively. The results are obtained from 30 independent runs assuming the maximum function evaluation number of 50,000 for all algorithms.

As shown in Table 5, for 10-dimensional problems, the Ad-TLBO finds the best solutions in most benchmarks. Except for the Quadric, Rosenbrock, and Ackley functions, this algorithm can converge to the globally optimal solution for all benchmarks. The algorithm converged to the globally optimal solution for the Rosenbrock function in 26 of 30 independent runs. Moreover, Table 6 shows that the proposed algorithm also outperforms the other algorithms in most benchmarks when the dimension is 30. Except for the Quadric, Zakharov, Ackley, and Rosenbrock functions, also in this challenge, the proposed algorithm can converge to the globally optimal solution for all benchmarks. For the Weierstrass and Griewank benchmarks, the original version of TLBO and its variants) except for ATLBO-DA(can also obtain the globally optimal solution.

Convergence rate: The mean number of function evaluations (*mFEs*) is used to compare the investigated algorithms in convergence rate. The *mFEs* value of the convergent algorithms to acceptable solutions in 30 independent runs is given in Table 7. The expression "NaN" indicates the non-convergence of the algorithm. The results in Table 7 indicate that Ad-TLBO converges to the acceptable solutions in the 10-dimensional function with the smaller mean number of function evaluations in all benchmarks, except for the Ackley, Quadric functions. Therefore, in most benchmarks, the convergence speed of the proposed algorithm is higher. For the Rosenbrock function, just Ad-TLBO, jDE, ABC, ATLBO-DA, and ELTLBO algorithms can converge to acceptable solutions. According to Table 8, the proposed algorithm also outperforms the other algorithms in all benchmarks in 30-dimensional functions (except Quadric). For Quadric and Zakharov benchmark, the original version of TLBO and its variants can also reach the acceptable solutions 1e-6. Except for Ad-TLBO and ATLBO-DA, all other algorithms cannot converge to the acceptable solutions for the 30-dimensional Rosenbrock function.

Reliability: The reliability characteristic of the algorithms under investigation is shown in Tables 7 and 8, with their success rate in converging to the acceptable solutions in 30 independent runs. These tables show that Ad-TLBO successfully reaches acceptable solutions for all benchmarks. Just Ad-TLBO, jDE, and ATLBO-DA can get the acceptable solutions with 100% success for the 10-dimensional

Table 9
Summary of the CEC'20 Test Suite (Yue et al., 2019).

No	Functions	Characteristic	Global Optimum
1	Shifted and Rotated Bent Cigar Function (CEC 2017 (Awad et al., 2016) F1)	Unimodal Function	100
2	Shifted and Rotated Schwefel's Function (CEC 2014 (Liang et al., 2014) F11)	Basic Functions	1100
3	Shifted and Rotated Lunacek bi-Rastrigin Function (CEC 2017 (Awad et al., 2016) F7)		700
4	Expanded Rosenbrock's plus Griewangk's Function (CEC2017 (Awad et al., 2016) F19)		1900
5	Hybrid Function 1 ($N = 3$) (CEC 2014 (Liang et al., 2014) F17)	Hybrid Functions	1700
6	Hybrid Function 2 ($N = 4$) (CEC 2017 (Awad et al., 2016) F16)		1600
7	Hybrid Function 3 ($N = 5$) (CEC 2014 (Liang et al., 2014) F21)		2100
8	Composition Function 1 ($N = 3$) (CEC 2017 (Awad et al., 2016) F22)	Composition Functions	2200
9	Composition Function 2 ($N = 4$) (CEC 2017 (Awad et al., 2016) F24)		2400
10	Composition Function 3 ($N = 5$) (CEC 2017 (Awad et al., 2016) F25)		2500
Search range $= [-100\ 100]^D,\ D = 5,\ 10$ *and* 15			

Table 10
Top-ranked optimization algorithms at the latest IEEE CEC competitions.

Method	Rank	Year
IMODE (JSallam et al., 2020)	1	2020
AGSK (Mohamed et al., 2020)	2	2020
HSES (Zhang & Shi, 2018)	1	2018
EBOwithCMAR (Kumar et al., 2017)	1	2017
LSHADE-cpEpSin (Awad et al., 2017)	3	2017
LSHADE-SPACMA (Mohamed et al., 2017)	4	2017

Rosenbrock function. The PSOwFIPS and CLPSO algorithms failed to converge acceptable solutions in any 30-dimensional benchmarks functions. Except for Ad-TLBO, none of the algorithms are entirely successful to convergence for all benchmarks, and for at least one of the benchmarks fail, or they are not 100% successful.

Statistical analysis: Nonparametric statistical procedures of the Quade test and Friedman test are used for statistical analysis of the results of challenge 2 in two states of $D = 10$ and $D = 30$. Figs. 9 and 10 show the average ranks of optimization algorithms in this challenge based on the standard errors of means. As it is clear from these figures, both tests introduce the proposed algorithm as the best algorithm in both 10- and 30-dimensional states.

Challenge3

In this section, algorithm performance is slightly more challenged using the last CEC competition in 2020 (Yue et al., 2019). CEC benchmarks competition is considered the most attractive competition suite among optimization problems (Mohamed et al., 2020). CEC 2020 consists of ten new hybrid and composite sets of functions presented in Table 9 (The F6 and F7 functions were eliminated from the competition for $D = 5$). *maxFEs* (The maximum number of function evaluations) is 50,000, 1,000,000 and 3,000,000 for $D = 5$, 10 and 15, respectively. Reach to *maxFEs* value or error value of 1e-8 are the termination criteria of the algorithm. Each benchmark runs 30 times independently. More details about this competition are presented in Ref. Yue et al. (2019).

The performance efficiency of the proposed Ad-TLBO algorithm is demonstrated by comparing its results with six state-of-the-art top-ranked optimization algorithms at the latest IEEE CEC competitions and other eight optimization algorithms that developed in recent years. The CEC's top-ranked algorithms, rankings, and year of competition are listed in Table 10.

AO (Abualigah et al., 2021), ATLBO-DA (Bureerat & Sleesongsom, 2021), AHA (Zhao et al., 2022), EO (Faramarzi et al., 2020b), MPA (Faramarzi et al., 2020a), QANA (Zamani et al., 2021), SMA (Li et al., 2020) and RUN (Ahmadianfar et al., 2021) are eight developed algorithms in recent years that have been used to compare with the proposed method. The results of Ad-TLBO and the other state-of-the-art algorithms are summarized in Tables 11–13. Due to the complication of CEC's benchmarks, the proposed Classified Search approach is activated in this challenge. For $D = 5$, the coefficients of K1 and K2, and K3 are selected by 0.35, 0.4, and 0.25, respectively. The initial population number in three sections is 25, 50, and 125 members, and $nc = 5$ is assumed. For $D = 10$, the coefficients of K1, K2, and K3 are selected by 0.55, 0.35, and 0.1, respectively. The initial population number in the three sections are 20, 200, and 400 members, and $nc = 25$ is assumed.

As shown in Table 11, for $D = 5$, the proposed method ranks one to three in five of the eight functions. In F10, the Ad-TLBO method is ranked one and better than the IMODE (rank 1 in CEC 2020). In F1, this rank is similar to other methods. In functions F2, F4, and F5, the proposed method could not be among the top ranks, but it was able to get an acceptable rank among these 15 algorithms and better than methods such as the HSES method (rank 1 in CEC2018). In the F4, the difference between the mean value of the Ad-TLBO and the AGSK (rank 2 in CEC2020) is minimal (about 0.05), and by repeating the runs, a different result may be obtained, and the ranking of the proposed method will be 5. In this function, the Ad-TLBO reached the global optimum solution (best value=0), while some better rank methods such as the AGSK and LSHADE-cpEpSin could not. As it is clear from the results of Table 12, by increasing the number of variables ($D = 10$), the efficiency of the proposed method is improved. In five functions (F1, F3, F6, F9, F10), the Ad-TLBO method ranks one and reaches even better results than the top-ranked algorithms such as IMODE and AGSK (rank 1 and 2 in CEC2020). In F1, this rank is similar to other methods. In F6, the proposed method, QANA and MPA reached the global optimum in all runs (mean and std values=0), while none of the other algorithms reached the global optimum even in one of the 30 independent runs. In this function, a maximum accuracy of 1e-6 is obtained by other algorithms, which is a considerable distance up to 1e-8. In F8, the rank of method Ad-TLBO is equal to 2. In this function, the proposed method performed better than all other methods except IMODE. In this function, the quality of the three Ad-TLBO, IMODE, and AGSK algorithms are much better than others. All algorithms get trapped in local optima in this function, except these three algorithms. In F4 and F5, the proposed algorithm did not perform well compared to the top-rank algorithms. However, its overall performance was acceptable among 15 algorithms, and it got a better answer than most of these algorithms, including HSES (rank1 in CEC2018). In general, the Ad-TLBO method in the $D = 10$ problem in six functions has a very good performance and in three maintain functions, the results are acceptable. As shown in Table 13 for $D = 15$, the Ad-TLBO method outperforms all the recently published methods (except MPA in F5 and AHA in F4) in almost all benchmarks. Compared with top-ranked CEC competition algorithms, the proposed method works well in F3, F6, F8, and F9 benchmarks, and in F1 and F10 benchmarks, it has obtained a joint first rank with other top-ranked algorithms. In F6, the Ad-TLBO reached the global optimum in all runs, while none of the other top-ranked algorithms reached the global optimum even in one of the 30 independent runs (maximum reached accuracy is 1e-2). In F9, the

Table 11
Comparison results of the Ad-TLBO with CEC's top-ranked algorithms and recently developed algorithms for $D = 5$. (Results of top-ranked CEC competitions algorithms are taken from Mohamed et al. (2020) and JSallam et al. (2020).

			Dimensions=5							
			F1	F2	F3	F4	F5	F8	F9	F10
Top-ranked CEC Competitions Algorithms	**LSHADE-cpEpSin**	**Mean**	**0.00E+00**	**2.98E+00**	**1.30E+00**	**6.88E-02**	**1.40E-01**	**3.14E-01**	**3.33E+00**	**3.02E+02**
		STD	**0.00E+00**	**2.27E+00**	**6.96E-01**	**3.90E-02**	**3.66E-01**	**1.72E+00**	**1.83E+01**	**8.65E+00**
		Best	**0.00E+00**	**1.25E-01**	**1.03E-01**	**3.53E-07**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**3.00E+02**
	EBOwithCMAR	**Mean**	**0.00E+00**	**1.54E-01**	**5.18E+00**	**6.07E-02**	**1.41E-01**	**1.21E+01**	**9.67E+01**	**3.37E+02**
		STD	**0.00E+00**	**1.02E-01**	**8.35E-02**	**2.54E-02**	**3.89E-01**	**3.08E+01**	**1.83E+01**	**4.57E+01**
		Best	**0.00E+00**	**0.00E+00**	**5.15E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1.00E+02**
	AGSK	**Mean**	**0.00E+00**	**1.64E+01**	**2.87E+00**	**1.11E-01**	**0.00E+00**	**0.00E+00**	**3.33E+01**	**2.25E+02**
		STD	**0.00E+00**	**2.58E+01**	**2.05E+00**	**6.05E-02**	**0.00E+00**	**0.00E+00**	**4.79E+01**	**1.32E+02**
		Best	**0.00E+00**	**6.14E-01**	**4.38E-07**	**1.67E-03**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	IMODE	**Mean**	**0.00E+00**	**8.33E-02**	**5.15E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.43E+02**
		STD	**0.00E+00**	**8.88E-02**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1.36E+02**
		Best	**0.00E+00**	**0.00E+00**	**5.15E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	LSHADE-SPACMA	**Mean**	**0.00E+00**	**4.42E-01**	**5.25E+00**	**9.81E-06**	**2.08E-02**	**0.00E+00**	**9.67E+01**	**3.47E+02**
		STD	**0.00E+00**	**1.24E+00**	**1.43E-01**	**5.37E-05**	**1.14E-01**	**0.00E+00**	**1.83E+01**	**5.23E-04**
		Best	**0.00E+00**	**6.90E-03**	**5.15E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**9.67E+01**	**3.47E+02**
	HSES	**Mean**	**0.00E+00**	**4.76E+01**	**5.41E+00**	**2.57E-01**	**3.32E+00**	**4.76E+01**	**1.00E+02**	**3.47E+02**
		STD	**0.00E+00**	**5.91E+01**	**1.97E-01**	**1.17E-01**	**2.12E+00**	**3.84E+01**	**2.52E-11**	**1.17E-02**
		Best	**0.00E+00**	**0.00E+00**	**5.15E+00**	**1.09E-01**	**9.95E-01**	**0.00E+00**	**1.00E+02**	**3.47E+02**
Recently developed Algorithms	**AO**	**Mean**	**1.50E+07**	**4.16E+02**	**2.01E+01**	**2.52E+00**	**1.59E+03**	**5.55E+01**	**1.18E+02**	**3.52E+02**
		STD	**1.12E+07**	**1.31E+02**	**5.58E+00**	**7.01E-01**	**1.78E+03**	**3.68E+01**	**2.90E+01**	**5.16E+01**
		Best	**1.93E+06**	**1.91E+02**	**1.23E+01**	**1.17E+00**	**1.06E+02**	**1.85E+01**	**1.03E+02**	**3.48E+02**
	ATLBO-DA	**Mean**	**1.89E+03**	**1.80E+02**	**8.20E+00**	**3.27E-01**	**5.92E+01**	**4.59E+01**	**9.48E+01**	**3.43E+02**
		STD	**3.21E+03**	**1.54E+02**	**2.59E+00**	**1.73E-01**	**6.93E+01**	**4.51E+01**	**2.17E+01**	**1.42E+01**
		Best	**0.00E+00**	**6.84E+00**	**5.79E+00**	**1.99E-02**	**3.11E-05**	**7.26E-07**	**3.64E-03**	**3.00E+02**
	AHA	**Mean**	**0.00E+00**	**6.28E+01**	**8.66E+00**	**2.40E-01**	**1.83E+01**	**1.73E+01**	**8.67E+01**	**3.36E+02**
		STD	**0.00E+00**	**6.81E+01**	**2.06E+00**	**1.44E-01**	**1.71E+00**	**3.82E+01**	**3.46E+01**	**4.61E+01**
		Best	**0.00E+00**	**1.25E-01**	**5.74E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1.00E+02**
	EO	**Mean**	**7.15E+01**	**1.06E+02**	**6.69E+00**	**2.45E-01**	**4.62E+01**	**5.13E+01**	**1.93E+02**	**3.46E+02**
		STD	**1.14E+02**	**7.90E+01**	**1.38E+00**	**7.76E-02**	**5.91E+01**	**5.02E+01**	**1.06E+02**	**8.57E+00**
		Best	**3.30E-01**	**3.80E-01**	**5.15E+00**	**1.10E-01**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**3.00E+02**
	MPA	**Mean**	**0.00E+00**	**1.51E+02**	**6.52E+00**	**3.20E-01**	**1.30E-02**	**1.68E+01**	**5.67E+01**	**3.00E+02**
		STD	**0.00E+00**	**4.31E+01**	**1.16E+00**	**1.26E-01**	**3.85E-02**	**3.82E+01**	**5.04E+01**	**8.26E+01**
		Best	**0.00E+00**	**7.60E-01**	**1.89E+00**	**1.11E-04**	**2.76E-08**	**0.00E+00**	**3.66E-07**	**1.00E+02**
	QANA	**Mean**	**0.00E+00**	**3.97E+01**	**6.41E+00**	**2.26E-01**	**9.83E-01**	**6.23E+00**	**1.00E+00**	**3.47E+02**
		STD	**0.00E+00**	**6.28E+01**	**1.00E+00**	**9.77E-02**	**9.42E-01**	**1.85E+01**	**0.00E+00**	**8.99E-03**
		Best	**0.00E+00**	**1.25E-01**	**5.31E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1.00E+02**	**3.47E+02**
	SMA	**Mean**	**2.49E+07**	**1.21E+02**	**3.78E+01**	**2.50E+00**	**1.24E+02**	**3.69E+01**	**1.97E+02**	**3.59E+02**
		STD	**1.77E+07**	**1.95E+01**	**1.04E+01**	**8.36E-01**	**6.27E+01**	**7.33E+00**	**4.72E+01**	**8.61E+00**
		Best	**1.57E+06**	**9.23E+01**	**1.81E+01**	**8.71E-01**	**4.57E+01**	**1.59E+01**	**1.06E+02**	**3.49E+02**
	RUN	**Mean**	**4.70E+03**	**8.08E+01**	**1.41E+01**	**6.20E-01**	**4.19E+01**	**3.12E+01**	**9.70E+01**	**3.47E+02**
		STD	**2.47E+03**	**6.80E+01**	**3.60E+00**	**3.34E-01**	**3.30E+01**	**3.91E+01**	**1.77E+01**	**5.62E-02**
		Best	**2.39E+03**	**6.90E+00**	**6.53E+00**	**2.49E-01**	**1.11E+01**	**3.44E-02**	**2.96E+00**	**3.47E+02**
Proposed Algorithm	**Ad-TLBO**	**Mean**	**0.00E+00**	**2.39E+01**	**3.30E+00**	**1.65E-01**	**2.33E+00**	**1.90E+00**	**2.33E+01**	**1.35E+02**
		STD	**0.00E+00**	**2.55E+01**	**2.74E+00**	**1.17E-01**	**2.47E+00**	**4.12E+00**	**4.30E+01**	**8.40E+01**
		Best	**0.00E+00**	**1.24E-01**	**7.12E-03**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
Rank			**1**	**6**	**2**	**6**	**8**	**3**	**3**	**1**

Table 12
Comparison results of the Ad-TLBO with CEC's top-ranked algorithms and recently developed algorithms for $D = 10$. (Results of top-ranked CEC competitions algorithms are taken from Mohamed et al. (2020) and JSallam et al. (2020).

			Dimensions=10								
			F1	F2	F3	F4	F5	F6	F8	F9	F10
Top-ranked CEC Competitions Algorithms	**LSHADE-cpEpSin**	**Mean**	**0.00E+00**	**3.51E+00**	**1.16E+01**	**1.46E-01**	**2.79E+01**	**3.32E-01**	**1.00E+02**	**3.5E+02**	**4.22E+02**
		STD	**0.00E+00**	**3.24E+00**	**4.63E-01**	**2.62E-02**	**4.47E+01**	**2.29E-01**	**1.15E-13**	**6.96E+01**	**2.32E+01**
		Best	**0.00E+00**	**1.25E-01**	**1.06E+01**	**9.86E-02**	**2.08E-01**	**1.15E-02**	**1.00E02**	**1.00E+02**	**3.98E+02**
	EBOwithCMAR	**Mean**	**0.00E+00**	**4.63E+00**	**1.04E+01**	**1.33E-01**	**1.01E+01**	**1.29E-01**	**1.00E+02**	**1.60E+02**	**4.10E+02**
		STD	**0.00E+00**	**3.80E+00**	**5.21E-02**	**2.50E-02**	**1.10E+01**	**9.11E-02**	**0.00E+00**	**9.80E+01**	**2.04E+01**
		Best	**0.00E+00**	**2.50E-01**	**1.04E+01**	**9.86E-02**	**2.08E-01**	**4.66E-03**	**1.00E+02**	**1.00E+02**	**3.98E+02**
	AGSK	**Mean**	**0.00E+00**	**2.84E+01**	**9.93E+00**	**5.83E-02**	**3.18E-01**	**1.55E-01**	**1.80E+01**	**7.63E+01**	**2.98E+02**
		STD	**0.00E+00**	**3.21E+01**	**4.26E+00**	**3.11E-02**	**3.06E-01**	**1.17E-01**	**2.38E+01**	**4.29E+01**	**1.43E+02**
		Best	**0.00E+00**	**4.09E+00**	**6.12E-01**	**1.94E-03**	**0.00E+00**	**2.20E-02**	**0.00E+00**	**0.00E+00**	**1.00E+02**
	IMODE	**Mean**	**0.00E+00**	**4.19E+00**	**1.21E+01**	**0.00E+00**	**3.88E-01**	**9.14E-02**	**2.72E+00**	**4.10E+01**	**3.98E+02**
		STD	**0.00E+00**	**3.70E+00**	**7.82E-01**	**0.00E+00**	**3.83E-01**	**5.08E-02**	**7.46E+00**	**4.46E+01**	**2.89E-13**
		Best	**0.00E+00**	**1.25E-01**	**1.07E+01**	**0.00E+00**	**4.03E-06**	**2.67E-02**	**0.00E+00**	**0.00E+00**	**3.98E+02**
	LSHADE-SPACMA	**Mean**	**0.00E+00**	**5.20E+00**	**1.29E+01**	**1.44E-01**	**2.25E+00**	**3.80E-01**	**1.00E+02**	**2.87E+02**	**4.10E+02**
		STD	**0.00E+00**	**9.39E+00**	**1.87E+00**	**1.70E-01**	**5.22E+00**	**1.94E-01**	**0.00E+00**	**9.53E+01**	**2.05E+01**
		Best	**0.00E+00**	**1.89E-01**	**1.06E+01**	**0.00E+00**	**0.00E+00**	**2.30E-02**	**1.00E+02**	**1.00E+02**	**3.98E+02**
	HSES	**Mean**	**0.00E+00**	**8.50E+00**	**1.41E+01**	**9.65E-01**	**1.03E+02**	**8.63E+00**	**1.00E+02**	**3.29E+02**	**4.46E+02**
		STD	**0.00E+00**	**4.08E+00**	**6.57E-01**	**2.59E-01**	**1.99E+02**	**3.00E+01**	**0.00E+00**	**1.09E+00**	**1.23E+00**
		Best	**0.00E+00**	**3.54E+00**	**1.04E+01**	**5.04E-01**	**2.09E-01**	**2.83E-01**	**1.00E+02**	**3.27E+02**	**4.44E+02**
Recently developed Algorithms	**AO**	**Mean**	**7.13E+08**	**1.20E+03**	**7.54E+01**	**3.63E+01**	**1.63E+04**	**9.54E-01**	**1.45E+02**	**3.50E+02**	**4.66E+02**
		STD	**3.58E+08**	**2.50E+02**	**1.16E+01**	**2.72E+01**	**1.03E+04**	**2.40E-01**	**1.32E+01**	**9.98E+01**	**2.94E+01**
		Best	**1.91E+08**	**5.48E+02**	**5.60E+01**	**1.09E+01**	**1.32E+03**	**5.46E-01**	**1.20E+02**	**1.50E+02**	**4.29E+02**
	ATLBO-DA	**Mean**	**1.85E+08**	**7.12E+02**	**4.03E+01**	**5.34E+01**	**7.05E+02**	**6.30E-01**	**1.26E+02**	**2.12E+02**	**4.24E+02**
		STD	**5.20E+08**	**2.70E+02**	**1.27E+01**	**1.77E+02**	**1.39E+03**	**2.35E-01**	**2.10E+01**	**1.27E+02**	**6.68E+01**
		Best	**2.67E+03**	**1.22E+02**	**1.71E+01**	**1.11E+00**	**3.75E+01**	**2.68E-01**	**7.28E+01**	**1.00E+02**	**1.03E+02**
	AHA	**Mean**	**0.00E+00**	**2.33E+02**	**2.41E+01**	**8.20E-01**	**5.76E+01**	**6.10E-01**	**9.86E+01**	**1.55E+02**	**4.25E+02**
		STD	**0.00E+00**	**1.38E+02**	**6.49E+00**	**3.30E-01**	**6.72E+01**	**1.62E-01**	**1.47E+01**	**9.87E+01**	**2.30E+01**
		Best	**0.00E+00**	**6.83E+00**	**1.35E+01**	**2.05E-01**	**1.41E+00**	**5.00E-01**	**2.06E+01**	**1.00E+02**	**3.98E+02**
	EO	**Mean**	**9.07E+02**	**2.70E+02**	**1.81E+01**	**7.73E-01**	**3.97E+02**	**8.33E-02**	**1.00E+02**	**3.43E+02**	**4.33E+02**
		STD	**1.19E+03**	**2.11E+02**	**4.32E+00**	**3.33E-01**	**2.28E+02**	**1.40E-01**	**2.95E-01**	**8.24E+00**	**2.10E+01**
		Best	**2.07E-01**	**7.13E+00**	**1.22E+01**	**1.97E-02**	**5.56E+01**	**4.64E-06**	**1.00E+02**	**3.30E+02**	**3.98E+02**
	MPA	**Mean**	**0.00E+00**	**6.46E+01**	**1.69E+01**	**5.14E-01**	**2.66E+00**	**0.00E+00**	**9.50E+01**	**1.18E+02**	**3.99E+02**
		STD	**0.00E+00**	**7.00E+01**	**2.22E+00**	**1.70E-01**	**4.27E+00**	**0.00E+00**	**2.12E+01**	**5.07E+01**	**8.33E+00**
		Best	**0.00E+00**	**1.87E-01**	**1.27E+01**	**2.07E-01**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1.00E+02**	**3.98E+02**
	QANA	**Mean**	**0.00E+00**	**1.62E+02**	**1.88E+01**	**7.92E-01**	**5.72E+01**	**0.00E+00**	**9.65E+01**	**2.37E+02**	**4.28E+02**
		STD	**0.00E+00**	**1.20E+02**	**3.55E+00**	**3.33E-01**	**5.42E+01**	**0.00E+00**	**2.00E+01**	**1.22E+02**	**2.26E+01**
		Best	**0.00E+00**	**6.95E+00**	**1.30E+01**	**3.45E-01**	**6.24E-01**	**0.00E+00**	**2.18E+01**	**1.00E+02**	**3.98E+02**
	SMA	**Mean**	**7.00E+08**	**9.72E+02**	**1.35E+02**	**1.06E+01**	**1.49e+04**	**4.60E-01**	**2.16E+02**	**3.70E+02**	**4.91E+02**
		STD	**2.88E+08**	**1.33E+02**	**1.73E+01**	**2.45E+00**	**5.27E+03**	**1.25E-01**	**1.99E+02**	**4.36E+00**	**1.79E+01**
		Best	**1.85E+08**	**5.73E+02**	**1.05E+02**	**6.88E+00**	**2.12E+03**	**7.27E-02**	**8.78E+01**	**3.63E+02**	**4.39E+02**
	RUN	**Mean**	**4.22E+03**	**2.53E+02**	**4.49E+01**	**9.67E-01**	**1.10E+03**	**3.36E-01**	**1.02E+02**	**1.80E+02**	**4.19E+02**
		STD	**1.85E+03**	**1.93E+02**	**1.06E+01**	**3.68E-01**	**4.25E+02**	**1.63E-01**	**7.65E-01**	**1.15E+02**	**2.30E+01**
		Best	**2.21E+02**	**1.03E+01**	**2.63E+01**	**4.85E-01**	**3.79E+02**	**1.89E-04**	**1.00E+02**	**1.00E+02**	**3.98E+02**
Proposed Algorithm	**Ad-TLBO**	**Mean**	**0.00E+00**	**4.74E+01**	**8.51E+00**	**6.79E-01**	**9.20E+01**	**0.00E+00**	**1.37E+01**	**2.70E+01**	**1.79E+02**
		STD	**0.00E+00**	**4.52E+01**	**3.03E+00**	**2.48E-01**	**5.95E+01**	**0.00E+00**	**9.34E+00**	**4.57E+01**	**1.03E+02**
		Best	**0.00E+00**	**3.78E+00**	**3.70E+00**	**9.90E-02**	**1.06E+01**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1.00E+02**
Rank			**1**	**7**	**1**	**7**	**9**	**1**	**2**	**1**	**1**

Table 13
Comparison results of the Ad-TLBO with CEC's top-ranked algorithms and recently developed algorithms for $D = 15$. (Results of top-ranked CEC competitions algorithms are taken from Mohamed et al. (2020) and JSallam et al. (2020).

			Dimensions=15								
			F1	F2	F3	F4	F5	F6	F8	F9	F10
Top-ranked CEC Competitions Algorithms	**LSHADE-cpEpSin**	**Mean**	0.00E+00	5.26E+01	1.00E+01	2.73E-01	2.41E+01	1.13E+00	9.11E+01	2.43E+02	4.00E+02
		STD	0.00E+00	5.02E+01	2.44E+00	4.69E-02	7.74E+00	5.40E-01	2.42E+01	1.11E+02	2.89E-13
		Best	0.00E+00	2.42E+00	1.83E+00	1.79E-01	1.15E+01	5.40E-01	2.15E-03	1.00E+02	4.00E+02
	EBOwithCMAR	**Mean**	0.00E+00	7.97E+00	1.56E+01	2.00E-01	2.79E+01	2.07E-01	1.00E+02	3.00E+02	4.00E+02
		STD	0.00E+00	7.77E+00	1.45E-02	2.26E-02	4.77E+01	1.19E-01	0.00E+00	1.23E+02	0.00E+00
		Best	0.00E+00	2.50E-01	1.56E+01	1.48E-01	3.12E-01	4.05E-02	1.00E+02	1.00E+02	4.00E+02
	AGSK	**Mean**	0.00E+00	1.85E+01	1.42E+01	1.42E-01	6.25E+00	4.02E-01	6.85E+01	9.67E+01	4.00E+02
		STD	0.00E+00	1.46E+01	4.27E+00	5.71E-02	4.32E+00	2.23E-01	3.85E+01	1.83E+01	2.60E-13
		Best	0.00E+00	3.12E+00	0.00E+00	4.74E-02	3.12E-01	1.72E-01	0.00E+0	0.00E+00	4.00E+02
	IMODE	**Mean**	0.00E+00	3.14E+00	1.61E+01	0.00E+00	7.79E+00	6.92E-01	4.18E+00	9.33E+01	4.00E+02
		STD	0.00E+00	3.22E+00	3.12E-01	0.00E+00	3.66E+00	2.52E-01	9.61E+00	2.54E+01	0.00E+00
		Best	0.00E+00	1.25E-01	1.567E+01	0.00E+00	1.15E+00	2.81E-01	0.00E+00	0.00E+00	4.00E+02
	LSHADE-SPACMA	**Mean**	0.00E+00	3.73E+00	1.61E+01	2.81E-01	7.01E+00	6.24E-01	1.00E+02	3.90E+02	4.00E+02
		STD	0.00E+00	3.38E+00	8.30E-01	2.23E-01	2.17E+01	2.76E-01	0.00E+00	1.39E+00	0.00E+00
		Best	0.00E+00	1.64E-01	1.56E+01	1.78E-01	1.56E-01	7.77E-02	1.00E+02	3.90E+02	4.00E+02
	HSES	**Mean**	0.00E+00	1.95E+02	1.79E+01	1.45E+00	4.22E+01	2.67E+01	1.00E+02	3.88E+02	4.00E+02
		STD	0.00E+00	1.26E+02	1.83E+00	3.07E-01	4.49E+01	5.70E+01	1.57E-13	1.93E+00	1.66E-10
		Best	0.00E+00	1.19E+02	1.61E+01	1.06E+00	8.96E+00	4.54E-01	1.00E+02	3.85E+02	4.00E+02
Recently developed Algorithms	**AO**	**Mean**	3.10E+09	2.29E+3	1.43E+02	4.96E+02	3.80E+05	0.00E+00	2.99E+02	5.16E+02	7.10E+02
		STD	8.54+08	2.59E+02	1.60E+01	9.29E+02	3.73E+05	0.00E+00	4.22E+01	6.30E+01	4.49E+01
		Best	1.33E+09	1.81E+03	1.09E+02	2.05E+01	5.67E04	0.00E+00	2.23E+02	2.14E+02	6.46E+02
	AHA	**Mean**	3.57E+03	3.92E+02	3.43E+01	1.11E+00	1.72E+02	4.82E+02	1.01E+02	3.88E+02	4.17E+02
		STD	5.38E+03	1.78E+02	8.14E+00	3.85E-01	9.36E+01	0.00E+00	6.61E-01	5.46E+01	6.36E+01
		Best	1.75E+02	1.26E+02	2.14E+01	3.76E-01	1.31E+01	4.82E+02	1.00E+02	1.00E+02	4.00E+02
	EO	**Mean**	6.76E+03	5.68E+02	3.24E+01	1.34E+00	7.10E+02	0.00E+00	1.68E+02	4.01E+02	4.00E+02
		STD	6.87E+03	2.47E+02	7.94E+00	3.39E-01	3.28E+02	0.00E+00	2.89E+02	4.79E+00	0.00E+00
		Best	1.30E+00	1.39E+02	1.84E+01	7.70E-01	1.94E+02	0.00E+00	1.00E+02	3.94E+02	4.00E+02
	MPA	**Mean**	9.98E+03	1.96E+02	2.60E+01	8.41E-01	3.99E+01	0.00E+00	9.50E+01	2.38E+02	4.00E+02
		STD	9.83E+03	1.17E+02	3.46E+00	2.36E-01	3.37E+01	0.00E+00	2.27E+01	1.43E+02	1.12E-01
		Best	4.85E+01	2.53E+00	2.05E+01	4.01E-01	8.21E+00	0.00E+00	2.15E+01	1.00E+02	4.00E+02
	QANA	**Mean**	0.00E+00	3.33E+02	3.11E+01	1.19E+00	1.93E+02	2.00E+02	9.53E+01	3.88E+02	4.00E+02
		STD	0.00E+00	2.10E+02	6.81E+00	3.96E-01	1.31E+02	0.00E+00	2.13E+01	5.47E+01	0.00E+00
		Best	0.00E+00	1.03E+01	1.87E+01	4.15E-01	8.27E+00	2.00E+02	0.00E+00	1.00E+02	4.00E+02
	SMA	**Mean**	7.48E+08	1.83E+03	1.72E+02	1.56E+01	4.20E+04	0.00E+00	1.63E+03	4.34E+02	6.81E+02
		STD	1.98E+08	1.73E+02	1.35E+01	2.85E+00	3.31E+04	0.00E+00	8.02E+02	6.51E+00	2.45E+01
		Best	3.59E+08	1.32E+03	1.47E+02	1.16E+01	4.73E+03	0.00E+00	2.16E+02	4.21E+02	6.16E+02
	RUN	**Mean**	2.14E+06	5.50E+02	9.83E+01	7.12E+00	1.28E+05	0.00E+00	1.10E+02	4.11E+02	4.92E+02
		STD	1.08E+06	2.30E+02	1.75E+01	1.53E+00	4.36E+04	0.00E+00	7.45E-01	8.14E+00	1.03E+02
		Best	5.83E+05	1.33E+02	5.83E+01	3.41E+00	5.77E+03	0.00E+00	1.08E+02	4.00E+02	4.05E+02
Proposed Algorithm	**Ad-TLBO**	**Mean**	0.00E+00	1.84E+02	9.56E+00	1.13E+00	1.45E+02	0.00E+00	1.85E+01	9.62E+01	4.00E+02
		STD	0.00E+00	1.19E+02	4.07E+00	1.64E+00	1.03E+02	0.00E+00	1.24E+01	2.69E+01	0.00E+00
		Best	0.00E+00	8.80E+00	2.98E+00	4.18E-01	2.49E+01	0.00E+00	0.00E+00	9.88E-02	4.00E+02
Rank			1	6	1	8	8	1	2	2	1

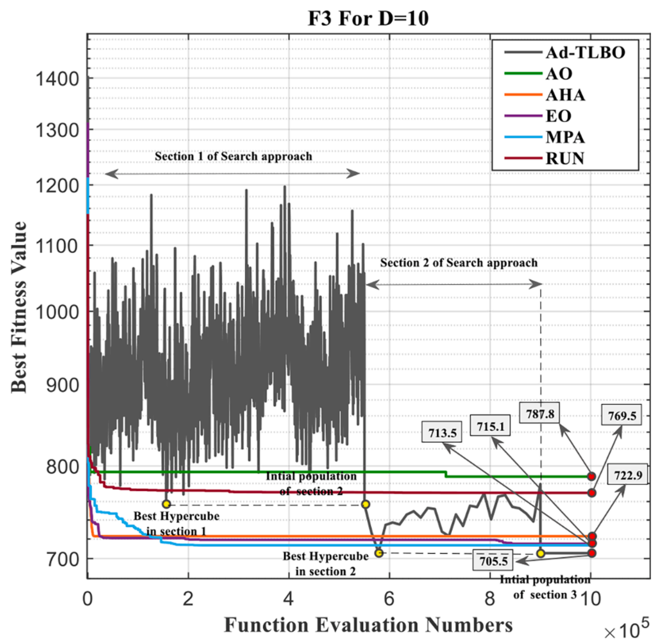

Fig. 11. Graphical comparison of some newly published algorithms with Ad-TLBO for F3.

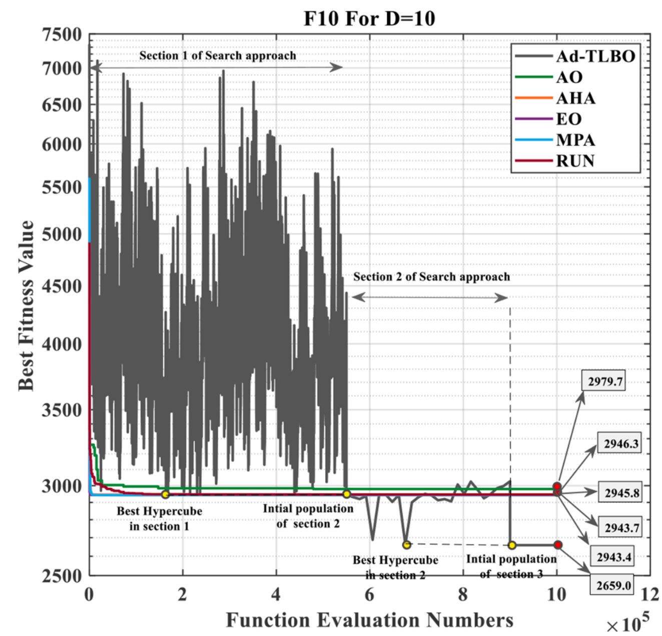

Fig. 13. Graphical comparison of some newly published algorithms with Ad-TLBO for F10.

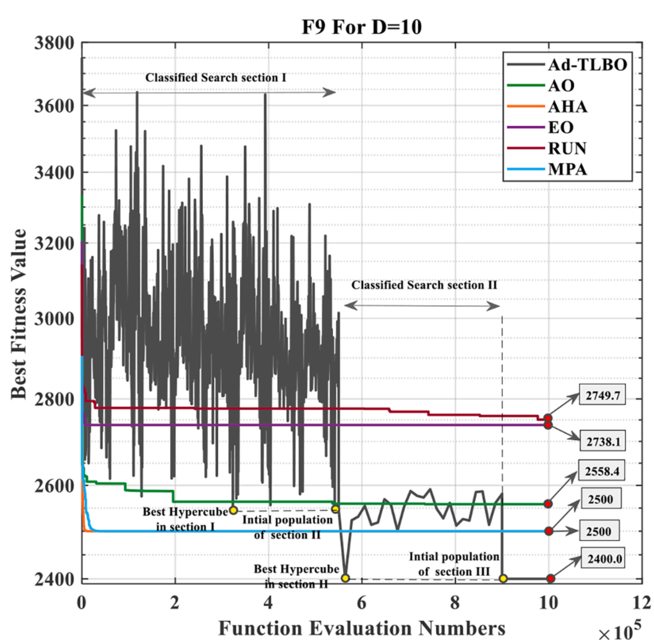

Fig. 12. Graphical comparison of some newly published algorithms with Ad-TLBO for F9.

proposed method is ranked two, with little difference from the IMODE algorithm. The results of this challenge show the competitive performance of the proposed method compared to 15 top-ranked and recently developed algorithms. The proposed method is among the top two in six of nine benchmarks.

Figs. 11–13 compare some newly published algorithms with the proposed algorithm in the 10-dimensional state for three benchmark functions F3, F9, and F10, graphically. Table 9 shows that the global optimum solution for functions F3, F9, and F10 is 700, 2400, and 2500, respectively. As shown in Figs. 11–13, the proposed algorithm has performed better than other optimization algorithms by a relatively good

Table 14
Comparison results of pressure vessels problem (the results of the algorithms before RUN algorithm are taken from Zhao et al. (2022)).

Algorithms	Mean	Best	Std	MFEs
GA2	6293.8432	6288.7445	7.4133	900,000
GA3	6177.2533	6059.9463	130.9297	80,000
CPSO	6147.1332	6061.0777	86.4500	240,000
HPSO	6099.9323	6059.7143	86.2000	81,000
PSO-DE	6059.7143	6059.7143	1.0E−10	42,100
PSO2	8756.6803	6693.7212	1492.5670	8000
QPSO	6440.3786	6059.7209	479.2671	8000
CDE	6085.2303	6059.7340	43.0130	204,800
CSA	6342.4990	6059.7140	250,000	250,000
ABC	6245.3080	6059.714339	205.0000	30,000
$(\mu + \lambda)ES$	6379.9380	6059.7016	210.0000	30,000
RUN	7389.5797	6335.1267	488.6675	30,000
AO	18,480.4178	10,473.5492	5142.2684	30,000
AHA	5885.7586	5885.3553	0.4142	30,000
ATLBO-DA	6076.4885	5885.3328	495.6848	30,000
SMA	8395.3778	6383.1629	1439.4226	30,000
MPA	5899.3200	5885.3687	50.4918	30,000
Ad-TLBO	5885.3911	5885.3334	0.1360	30,000
QANA	5972.6559	5887.60970	83.5602	30,000
EO	6412.6802	5919.1025	507.4803	30,000
Rank	1			

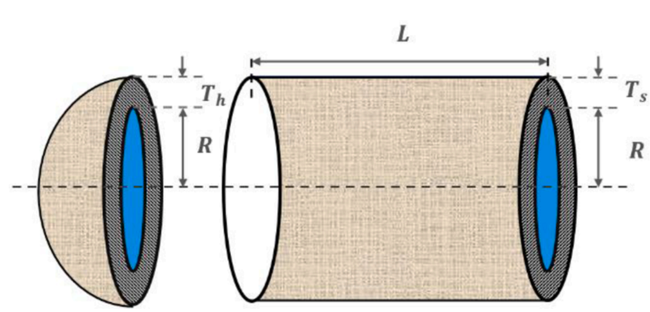

Fig. 14. Pressure vessels problem.

Table 15
Algorithm complexity.

D	T0 (s)	T1(s)	T2(s)	(T2 − T1)/T0
5	4E-02	4.93E-01	5.18E+00	1.17E+02
10		5.69E-01	5.28E+00	1.18E+02
15		1.05E+00	5.94E+00	1.22E+02

margin. The search technique proposed in this manuscript is quite evident in these figures. In the first two parts of the search technique, by dividing the search space into smaller hypercubes, the coverage of the search domain in the space has been improved. The cause of the fluctuation of the graph in the first two sections is a segmented search. In the first section of the search, the search space is segmented into 2^{nVar} smaller cubes, and the search continues in each cube. This search is done in 1024 small cubes for a ten variables problem, so the solutions fluctuate 1024 times with the increase in the number of function evaluations. In the second section of the search, the exploration continues only in the "*nc*" candidate hypercube, so the solutions have fluctuated for the "*nc*" time. Finally, in the last part of the search, since only the search was done in a hypercube, the solutions decrease with the increase of function evaluations number, like other algorithms.

Challenge4: engineering problem

Pressure vessels are widely used in various industries, including the aerospace industry. Launch vehicles or satellite orbital transfer systems use these structures as propellant storage tanks for the propulsion system. In this challenge, the minimum fabrication cost of these pressure vessels is considered the objective function, along with four inequality constraints. This problem has already been investigated in references such as Askarzadeh (2016), (Rao et al., 2011; Coello, 2000; Coello & Montes, 2002; dos Santos Coelho, 2010; He & Wang, 2007b, 2007a; Huang et al., 2007; Liu et al., 2010; Mezura-Montes & Coello, 2005), and Zhao et al. (2022) by many heuristic techniques. The results of them are summarized in Table 14.

In this section, in addition to the mentioned references, this problem has been investigated with eight optimization methods that have been published recently. The number of 30,000 function evaluations has been chosen as the stopping criteria for these algorithms. The evaluation criteria are the mean, standard deviation, and best fitness values obtained from 30 independent runs. The design variables are shown in Fig. 14. This optimization problem is described below:

$$
\begin{aligned}
&\min_x F = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3\\
&x = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]\\
&\boldsymbol{s.t},\ \ g_1(x) = -x_1 + 0.0193x_3 \le 0\\
&g_2(x) = -x_2 + 0.0095x_3 \le 0\\
&g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0\\
&g_4(x) = -x_4 - 240 \le 0\\
&0 \le x_1 \le 99,\ \ 0 \le x_2 \le 99,\ \ 10 \le x_3 \le 200,\ \ 10 \le x_4 \le 200,
\end{aligned}
\tag{19}
$$

As shown in Table 14, the proposed method was able to overcome other algorithms and achieve acceptable results in terms of the mean value, standard deviation, and the best fitness value. Also, compared to the first few methods in the table, these results have been obtained in fewer evaluations.

Algorithm complexity

In order to evaluate the computational complexity of the proposed algorithm, the method described in the CEC 2020 competition (Yue et al., 2019) has been used. According to this method, to calculate the computational complexity, it is necessary to calculate three times T0, T1, and T2. The following code calculates the execution time of T0:

$$
\begin{aligned}
&X = 0.55;\\
&\boldsymbol{For}\, \boldsymbol{i} = 1 : 1000000\\
&x = x + x;\ \ x = x/2;\ \ x = x * x;\ \ x = sqrt(x);\\
&x = log(x);\ \ x = exp(x);\ \ x = x/(x+2);\\
&\boldsymbol{end}
\end{aligned}
\tag{20}
$$

T1 represents the execution time of the F1 function for 200,000 evaluations in 5, 10 and 15 dimensions. The T2 execution time is obtained from the mean of 5 runs of the proposed algorithm on the F1 function for 200,000 evaluations. The results are summarized in Table 15.

Conclusions

The proposed advanced algorithm in the present study is based on the initial version of the TLBO algorithm. This advancement is done in two general parts. These modifications focused on improving the exploration and exploitation characteristics of O-TLBO. By modifying the algorithm's structure and the initialization process, we found that these steps improve the performance and efficiency of the algorithm significantly. To demonstrate this, four different challenges were considered to assess the performance of the proposed algorithms in terms of accuracy, convergence rate, and reliability. For comparison, various optimization algorithms were employed in several benchmark functions with variations in multimodal, separable, and differentiable characteristics in these challenges. In Challenge 1, the sphere benchmark was used to show the effect of modifying each step of part one compared to the O-TLBO or the steps before the modifications. We showed step-by-step improvement in the convergence rate and accuracy when the modifications were applied. In Challenge 2, this comparison was performed with eleven algorithms on nine benchmarks in terms of accuracy, convergence rate, and reliability in two modes of 10D and 30D functions. We illustrated that the proposed Ad-TLBO algorithm is the only algorithm that could achieve 100% success in all benchmarks.

Besides, the other algorithms, especially in the 30-dimensional cases, could not converge 100% to the acceptable value. Based on the results of this challenge, the proposed algorithm had a higher convergence speed and, at the same time, had much more accuracy compared to other algorithms. The last CEC competition in 2020 benchmark functions was used in Challenge 3 to demonstrate the efficacy of the proposed algorithm compared to the other investigated optimization algorithms. This competition consists of ten new hybrid and composite sets of functions. The performance efficiency of the proposed Ad-TLBO algorithm was demonstrated by comparing its results with six state-of-the-art top-ranked optimization algorithms at the latest IEEE CEC competitions and eight developed algorithms in recent years. As Challenge 3 indicates, increasing the number of variables improves the efficiency of the Ad-TLBO. In general, in this challenge, the proposed method had acceptable performance, especially in 10 and 15-dimensional functions, and in six out of nine benchmark functions, it was among the top two ranks. Finally, an engineering problem was investigated in the final challenge. The results of the fourth challenge show the quality of the answer of the proposed method compared to eight recently developed optimization algorithms in terms of the mean value, standard deviation, and maximum function evaluation.

The present study shows that the Ad-TLBO algorithm effectively finds the optimal global solution for various single-objective optimization problems. These results indicate that although the Ad-TLBO algorithm does not have entirely desirable performance, it can be improved by being studied further by researchers and extended to a broader range of application optimization problems. However, real-life optimization problems, especially engineering problems, often involve more than one objective. Therefore, the non-dominated sorting Ad-TLBO algorithm for multi-objective optimization and performance evaluation of this method is recommended for future studies. It should be noted that the hybridization of Ad-TLBO with other optimization algorithms or search techniques can improve the performance of this algorithm.

CRediT authorship contribution statement

Mohammad Fatehi: Conceptualization, Formal analysis, Validation, Investigation, Methodology, Writing – original draft, Visualization, Writing – review & editing, Software. **Alireza Toloei:** Conceptualization, Investigation, Writing – review & editing. **S.T.A. Niaki:** Conceptualization, Investigation, Writing – original draft, Visualization, Writing – review & editing. **Enrico Zio:** Conceptualization, Investigation, Visualization, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Abualigah, L., et al. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering, 15*, Article 107250.
Ahmadianfar, I., et al. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Systems with Applications, 18*, Article 115079.
Ahmadi-Nedushan, B., & Fathnejat, H. (2022). A modified teaching–learning optimization algorithm for structural damage detection using a novel damage index based on modal flexibility and strain energy under environmental variations. *Engineering with Computer, 38*, 847–874.
Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures, 16*, 1–12.
Awad, N. H., Ali, M. Z., Suganthan, P. N., Liang, J. J., & Qu, B. Y. (2016). *Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization*. Singapore: Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University. October.
Awad, N. H., Ali, M. Z., & Suganthan, P. N. (2017). Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In *Proceedings of the IEEE congress on evolutionary computation (CEC)*. IEEE.
Bui, Q. A. T., et al. (2022). Hybrid model: Teaching learning-based optimization of artificial neural network (TLBO-ANN) for the prediction of soil permeability coefficient. *Mathematical Problems in Engineering, 2022*, Article 8938836. https://doi.org/10.1155/2022/8938836
Bureerat, S., & Sleesongsom, S. (2021). Constraint handling technique for four-bar linkage path generation using self-adaptive teaching–learning-based optimization with a diversity archive. *Engineering Optimization, 53*, 513–530.
Chen, D., Lu, R., Zou, F., & Li, S. (2016a). Teaching-learning-based optimization with variable-population scheme and its application for ANN and global optimization. *Neurocomputing, 17*, 1096–1111.
Chen, D., et al. (2016b). Teaching-learning-based optimization with variable-population scheme and its application for ANN and global optimization. *Neurocomputing, 17*, 1096–1111.
Chen, Xu, et al. (2018). Teaching–learning–based artificial bee colony for solar photovoltaic parameter estimation. *Applied Energy, 21*, 1578–1588.
Coello, C. A. C (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry, 41*, 113–127.
Coello, C. A. C, & Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics, 16*, 193–203.
Das, S. P., & Padhy, S. (2018). A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting. *International Journal of Machine Learning and Cybernetics, 9*, 97–111.
Dib, F., & Boumhidi, I. (2017). Hybrid algorithm DE–TLBO for optimal H∞ and PID control for multi-machine power system. *International Journal of System Assurance Engineering and Management, 8*, 925–936.
dos Santos Coelho, L. (2010). Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Systems with Applications, 37*, 1676–1683.
Faramarzi, A., et al. (2020a). Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications, 15*, Article 113377.
Faramarzi, A., et al. (2020b). Equilibrium optimizer: A novel optimization algorithm. *Knowledge Based Systems, 19*, Article 105190.
He, Q., & Wang, L. (2007a). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation, 186*, 1407–1422.
He, Q., & Wang, L. (2007b). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence, 20*, 89–99.
Huang, F., Wang, L., & He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation, 186*, 340–356.
Ji, X., et al. (2017). An improved teaching-learning-based optimization algorithm and its application to a combinatorial optimization problem in foundry industry. *Applied Soft Computing, 5*, 504–516.
JSallam, K. M., et al. (2020). Improved multi-operator differential evolution algorithm for solving unconstrained problems. In *Proceedings of the IEEE congress on evolutionary computation (CEC)*. IEEE.
Kumar, A., Misra, R. K., & Singh, D. (2017). Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In *Proceedings of the IEEE congress on evolutionary computation (CEC)*. IEEE.
Kumar, S., et al. (2022). Multi-objective teaching-learning-based optimization for structure optimization. *Smart Science, 10*, 56–67.
Li, S., et al. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems, 11*, 300–323.
Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2014). *Problem definitions and evaluation criteria for the CEC 2014 special session and competition on real-parameter optimization*. Singapore: Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University. January.
Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing, 10*, 629–640.
Mezura-Montes, E., & Coello Coello, C. A. (2005). Useful infeasible solutions in engineering optimization with evolutionary algorithms. In *Proceedings of the Mexican international conference on artificial intelligence*. Springer.
Mohamed, A. W., et al. (2017). LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In *Proceedings of the IEEE congress on evolutionary computation (CEC)*. IEEE.
Mohamed, A. W., et al. (2020). Evaluating the performance of adaptive gaining sharing knowledge-based algorithm on CEC 2020 benchmark problems. In *Proceedings of the IEEE congress on evolutionary computation (CEC)*. IEEE.
Patel, J., et al. (2017). Layout optimization of a wind farm to maximize the power output using enhanced teaching learning based optimization technique. *Journal of Cleaner Production, 15*, 81–94.
Qu, X., et al. (2017). An improved TLBO based memetic algorithm for aerodynamic shape optimization. *Engineering Applications of Artificial Intelligence, 5*, 1–15.
Rao, R. V. (2016). *Teaching learning-based optimization algorithm* (pp. 9–39). Cham: Springer.
Rao, R. V., & Patel, V. (2013). An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *Scientia Iranica, 20*, 710–720.
Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer Aided Design, 43*, 303–315.
Rao, R., & Patel, V. (2012). An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations, 3*, 535–560.
Ren, Z., et al. (2022). A multi-objective elitist feedback teaching–learning-based optimization algorithm and its application. *Expert Systems with Applications, 18*, Article 115972.
Shao, W., Pi, D., & Shao, Z. (2017). An extended teaching-learning based optimization algorithm for solving no-wait flow shop scheduling problem. *Applied Soft Computing, 6*, 193–210.
Tang, Q., et al. (2017). Balancing stochastic two-sided assembly line with multiple constraints using hybrid teaching-learning-based optimization algorithm. *Computers & Operations Research, 8*, 102–113.
Wu, Di, et al. (2022). An improved teaching-learning-based optimization algorithm with reinforcement learning strategy for solving optimization problems. *Computational Intelligence and Neuroscience, 2022*, Article 1535957. https://doi.org/10.1155/2022/1535957
Yue, C. T., Price, K. V., Suganthan, P. N., Liang1, J. J., Ali, M. Z., Qu, B. Y., & Biswas, P. P. (2019). *Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization*. Tech. Rep., Zhengzhou University and Nanyang Technological University.
Zamani, H., Nadimi-Shahraki, M. H., & Gandomi, A. H. (2021). QANA: Quantum-based avian navigation optimizer algorithm. *Engineering Applications of Artificial Intelligence, 10*, Article 104314.
Zhang, G., & Shi, Y. (2018). Hybrid sampling evolution strategy for solving single objective bound constrained problems. In *Proceedings of the IEEE congress on evolutionary computation (CEC)*. IEEE.
Zhao, W, Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering, 38*, Article 114194.
Zou, F., Chen, D., & Xu, Q. (2019). A survey of teaching–learning-based optimization. *Neurocomputing, 33*, 366–383.