International Journal of Machine Learning and Cybernetics (2019) 10:1357–1371
https://doi.org/10.1007/s13042-018-0815-8

ORIGINAL ARTICLE

A modified teaching–learning-based optimization algorithm for numerical function optimization

Peifeng Niu[1] · **Yunpeng Ma**[1] · **Shanshan Yan**[2]

Received: 16 June 2017 / Accepted: 10 April 2018 / Published online: 17 April 2018

Abstract
In this paper, a kind of modified teaching–learning-based optimization algorithm (MTLBO) is proposed to enhance the solution quality and accelerate the convergence speed of the conventional TLBO. Compared with TLBO, the MTLBO algorithm possesses different updating mechanisms of the individual solution. In teacher phase of the MTLBO, the students are divided into two groups according to the mean result of learners in all subjects. Moreover, the two groups present different updating strategies of the solution. In learner phase, the students are still divided into two groups, where the first group includes the top half of the students and the second group contains the remaining students. The first group members increase their knowledge through interaction among themselves and study independently. The second group members increase their marks relying on their teacher. According to the above-mentioned updating mechanisms, the MTLBO can provide a good balance between the exploratory and exploitative capabilities. Performance of the proposed MTLBO algorithm is evaluated by 23 unconstrained numerical functions and 28 CEC2017 benchmark functions. Compared with TLBO and other several state-of-the-art optimization algorithms, the results indicate that the MTLBO shows better solution quality and faster convergence speed.

Keywords Teaching–learning-based optimization · Modified teaching–learning-based optimization · Exploratory and exploitative capabilities · Unconstrained numerical functions · CEC2017

1 Introduction

Many real-life optimization problems possess complicated properties, such as multimodality, high dimensionality and non-differentiability, so that they are difficult to solve. Many experts and scholars have indicated that exact optimization techniques, such as steepest decent, dynamic programming and linear programming, failed to provide an optimal solution for optimization problems of these types [1, 2]. For instance, many traditional optimization methods require the gradient information of optimization problems so that they cannot approach nondifferentiable problems. Therefore, a great deal of efficient nature-inspired meta-heuristic optimization techniques, which do not demand the gradient information of optimization problems, have been proposed to address these complex optimization problems.

During the last three decades, the research of meta-heuristics intelligent optimization algorithm has become a research hotspot. Many state-of-the-art swarm optimization algorithms were proposed and developed in recent decades. Particle swarm optimization (PSO) [3] was proposed based on the foraging behavior of birds. Artificial bee colony [4] was proposed based on the foraging behavior of honey bees. Gravitational search algorithm (GSA) [5] was proposed based on the principle of gravitational force. Krill herds (KH) algorithm [6] was proposed based on the foraging behavior of krill herds. Social-spider optimization algorithm (SSO) [7] was proposed based on the cooperative behavior between the female spiders and the male spiders. Sine Cosine Algorithm (SCA) [8] was proposed based on the sine and cosine functions. Teaching–learning-based optimization (TLBO) [9] algorithm was proposed based on the teaching–learning behavior of class. Moreover, these nature-inspired meta-heuristic optimization methods have been proved that they are suitable to solve those complex function problems [4, 10, 11] and difficult real-life problems [12–15].

✉ Yunpeng Ma
18232362368@163.com

[1] School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, Hebei, China

[2] Hydropower Station of Administration of Taolinkou Reservoir, Qinhuangdao 066004, Hebei, China

Teaching–learning-based optimization algorithm (TLBO) is a sort of novel population-based optimization method, which is proposed to obtain global solutions of continuous non-linear functions or engineering optimization problems. It has several superior properties, such as less computational effort, high consistency and less setting parameters. The TLBO has been applied to a wide range of real-world optimization problems, such as electrical engineering [16–19], manufacturing processes [20, 21] and economic load dispatch [22]. However, many researchers still proposed a large number of variants to improve the performance of TLBO algorithm. In order to improve the solution quality and quicken the convergence speed of the TLBO, Li et al. [23] proposed a kind of ameliorated teaching learning based optimization algorithm. In [24], the elitism mechanism is introduced in TLBO to enhance its performance. To enhance the exploration and exploitation capacities of TLBO, Rao et al., introduced some improved mechanisms in teaching–learning-based optimization algorithm [25]. In literature [26], quasi-opposition based learning concept was integrated with original TLBO to accelerate the convergence speed of TLBO.

In literature [1], Rao R.V. etc. indicated some nature-inspired population-based optimization algorithms had several limitations in one or the other aspects. Hence, many researchers proposed all kinds of mechanisms to improve the performance of algorithms. The enhancement of one algorithm is divided into two ways. The first way was done by modifying the existing algorithms. The second way was done by combining the strengths of different optimization algorithms. In this paper, a modified teaching–learning-based optimization algorithm is proposed by the first way.

Although the TLBO algorithm has shown good performance for a wide range of real-world optimization problems, its solution quality and convergence speed are still improved further by some methods. In order to improve the solution quality and convergence speed of TLBO, a modified teaching–learning-based optimization algorithm is proposed, namely MTLBO. In this paper, a new concept is firstly introduced, which is called 'actual teaching learning situation' (ATLS). In a real-life class, superior student has good self-learning ability and comprehensive ability, who either obtains knowledge from teacher and a more superior student or studies independently. For an underachiever, he gets knowledge from teacher or a more superior student principally. Moreover, a good teacher makes his or her efforts to bring the learners to his or her level in terms of knowledge. The MTLBO algorithm also has two phases, namely 'teacher phase' and 'learner phase'. In teacher phase, all students are divided into two groups according to the mean result in the class. We assume that if one student's comprehensive mark is higher than the mean result, the student is considered as a superior. Otherwise, the student is regarded as an underachiever. In learner phase, the students are also divided into two groups. The first group includes the top half of the student and the second group contains the remaining students. The first group members are regarded as superior students. Oppositely, the second group members are underachievers. According to the analysis of ATLS, a superior and an underachiever have different solution updating mechanisms. The detailed computation process of MTLBO is presented in Sect. 3. It is noted that the MTLBO algorithm has a good balance between exploration and exploitation. In literature [27], authors defined the exploration and exploitation, "*Exploration* is the process of visiting entirely new regions of a search space, whilst *exploitation* is the process of visiting those regions of a search space within the neighborhood of previously visited points". To evaluate the performance of MTLBO, 23 unconstraint benchmark numerical functions and 28 CEC2017 benchmark functions are selected as the test suite. Compared with the original TLBO, the proposed MTLBO algorithm not only enhances the solution quality, but also improves the convergence speed on most testing functions. In addition, simulations on 23 benchmark functions demonstrate that MTLBO outperforms several algorithms in general, including the artificial bee colony (ABC) [4], gravitational search algorithm(GSA) [5], Krill herds(KH) algorithm [6], social-spider optimization algorithm (SSO) [7], Sine Cosine Algorithm (SCA) [8].

The main contributions of this paper are summarized as follows:

1. This paper firstly proposes the concept of 'actual teaching learning situation' (ATLS).
2. Based on ATLS, a modified teaching–learning-based optimization algorithm is proposed.
3. The proposed MTLBO is applied to solve 23 unconstraint benchmark functions and 28 CEC2017 functions.
4. The solution quality and convergence speed of the original TLBO are improved obviously.

The rest of this paper is organized as follows. Section 2 reviews the basic TLBO algorithm. Section 3 presents the proposed MTLBO algorithm in detail. Section 4 shows the performance evaluation of the MTLBO. Section 5 concludes this paper.

2 Teaching–learning-based optimization algorithm

Teaching–learning-based optimization algorithm (TLBO) is a novel population-based meta-heuristic intelligent algorithm, which is inspired and proposed by the influence of a teacher on the output of learners in a class. For the TLBO algorithm, it has two vital parts, namely 'Teacher phase' and 'Learner phase'. The teaching–learning-based optimization algorithm is described briefly as follows.

2.1 Teacher phase

In teacher phase, learners obtain knowledge from their teacher. The teacher is regarded as the most knowledgeable person in a class, who makes big efforts to bring learners up to his or her level. Supposed that at any iteration i, M_i is the mean value of the marks and T_i is the teacher. The teacher will put effort to move the mean value M_i to its own level. However, it is hardly realized and the teacher is able to improve the mean of the class room depending on his or her capability. Therefore, the difference between the teacher T_i and the existing mean M_i is given by the Eq. (1). In this phase, the existing solution is updated based on the *diff* according to the following expression (2).

$$diff = r_i(T_i - T_F M_i) \tag{1}$$

$$X_{new,i} = X_{old,i} + diff \tag{2}$$

where $X_{old,i}$ is the ith learner's mark before updating, $X_{new,i}$ is the mark after learning from the teacher. r_i is the uniform random numbers from 0 to 1.$T_F = round[1 + rand(0, 1)\{2 - 1\}]$ is a teaching factor, which controls the mean value to be changed. During the algorithm, teaching factor is generated randomly in the range [1, 2]. If T_F is equal to 1, the mean value of the class room is not increased in the knowledge level. Oppositely, 2 indicate that the mean value of the class room completes transfer of knowledge.

2.2 Learner phase

In this phase, learners increase their knowledge through learning mutually. A learner can improve his or her knowledge through interacting randomly with other learners, such as group discussions, presentations and formal communications. Moreover, a learner gains knowledge from the more knowledge and experience person. The modification process of learners could be described as follows.

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

$$X_{new,i} = \begin{cases} X_{old,i} + r_i(X_i - X_j) & if\ f(X_i) < f(X_j) \\ X_{old,i} + r_i(X_j - X_i) & if\ f(X_i) > f(X_j) \end{cases} \tag{3}$$

The X_{new} is accepted if it gives a better function value.

3 The modified teaching–learning-based optimization algorithm

In this section, a new concept is firstly introduced, which is called 'actual teaching learning situation' (ATLS). In a real-life class, one student is considered as a superior student or an underachiever according to his or her comprehensive results. For a superior student, he or she has good self-learning ability, so he or she can not only increase his or her knowledge relying on self-study, but also obtain knowledge from teacher or a more superior student. For an underachiever, they obtain knowledge from their teacher or a more superior student principally. Moreover, a good teacher makes his or her efforts to bring the learners to his or her level in terms of knowledge. Based on the actual 'teaching learning' situation, a modified teaching–learning-based optimization algorithm, namely MTLBO, is proposed to enhance the solution quality and accelerate the convergence speed of the conventional TLBO. The MTLBO algorithm is described in detail as follows.

3.1 Teacher phase

In this phase, all students are divided into two groups based on the mean value of the class room. One group contains the superior students, another includes the underachievers. We assume that if one student's comprehensive mark is higher than the mean mark of the class, the student is considered as a superior student. Otherwise, the student is regarded as an underachiever. In this paper, the proposed MTLBO is used to solve the minimum optimization problem, so the individual presents the best performance with the lowest fitness value. In MTLBO, for a superior student, he can get knowledge from the best individual and study independently. The meaning of study independent is that the superior student still searches the optimal solution along with his previous information direction. For an underachiever, he obtains knowledge from his teacher. Based on the above analysis, the solution updating mechanism is shown in Eq. (4). Moreover, the expression of inertia weight is presented in Eq. (5).

$$X_{new,i} = \begin{cases} X_{old,i} \times W + (X_{best} - X_{old,i}) \times rand & if \quad f(X_{old,i}) < f(X_{mean}) \\ (X_{old,i} + (rand - 0.5) \times 2 \times (X_{mean} - X_{old,i})) \times \sin\left(\frac{\pi}{2} \times \frac{iter}{MaxIter}\right) + diff \times \cos\left(\frac{\pi}{2} \times \frac{iter}{MaxIter}\right) & if \quad f(X_{old,i}) > f(X_{mean}) \end{cases} \tag{4}$$

$$W = \omega_{start} - (\omega_{start} - \omega_{end}) \times \frac{iter}{MaxIter} \tag{5}$$

Seen from Eq. (4), X_{mean} is the mean result, W is the inertia weight which decides to balance the exploration and exploitation ability. Additionally, we also introduce $\sin(\frac{\pi}{2} \times \frac{iter}{MaxIter})$ and $\cos(\frac{\pi}{2} \times \frac{iter}{MaxIter})$ as inertia weights which can accelerate the convergence speed. *iter* is the current iteration, *MaxIter* is the maximum iteration. Seen from Eq. (5), the inertia weight descends linearly fromω_{start} to ω_{end}. Therefore, the adjustment process of inertia weight allows the MTLBO algorithm to explore the search space in the initial steps and to exploit the optimal solution in the latter steps.

It is noted that the two inertia weights $\sin(\frac{\pi}{2} \times \frac{iter}{MaxIter})$ and $\cos(\frac{\pi}{2} \times \frac{iter}{MaxIter})$ are firstly introduced to accelerate the convergence speed of TLBO. With the operation of MTLBO, the changing curves of two inertia weights are shown in Fig. 1. For an underachiever, his fitness value $f(X_{old,i})$ is higher than the mean fitness value $f(X_{mean})$, so he obtains knowledge from his teacher at the early stage. So the teacher plays an important role to improve the student's knowledge in the initial step. With the running of MTLBO, all individuals close to the optimal solution gradually if the algorithm shows good global convergence ability. Therefore, to avoid trapping into local optimum, the diversity of the population is increased in the latter step. Based on the analysis, the solution quality and convergence speed of MTLBO are indeed enhanced by the two inertia weights.

3.2 Learner phase

After teacher phase, the fitness values of all learners are sorted in ascending order. Then, the students are divided into two groups, where the first group includes the top half students and the second group contains the remaining students. The first group members are regarded as superior students, so they can not only obtain knowledge from a more superior student, but also study independently. The second group members get knowledge from their teacher principally. Therefore, the first group students update their results based on Eq. (6). On the contrary, the second group learners update their results according to Eq. (7).

$$\begin{aligned} &if\ f(X_{old,i}) > f(X_{neighbour}) \\ &\quad X_{new,i} = X_{old,i} + (X_{neighbour} - X_{old,i}) \times \cos\left(\frac{\pi}{2} \times \frac{iter}{MaxIter}\right) \\ &else \\ &\quad X_{new,i} = X_{old,i} + (rand - 0.5) \times 2 \times (X_{upper\lim it} - X_{lower\lim it}) \\ &end \end{aligned} \tag{6}$$

$$X_{new,i} = X_{old,i} + (X_{best} - X_{old,i}) \times \cos\left(\frac{\pi}{2} \times \frac{iter}{MaxIter}\right) \tag{7}$$

Shown in Eq. (6), in jth iteration, for a learner X_i, randomly selects a learner $X_{neighbour}$, where $neighbour \neq i$. If $X_{neighbour}$ has a smaller fitness value than X_i, the student X_i will obtain the knowledge from $X_{neighbour}$; otherwise, he will learn knowledge by himself. Based on this kind of mechanism, the diversity of the population will be increased and the convergence speed will be quickened simultaneously. For the second group members, there is a big gap between the latter of half learners and the teacher, so a big correction is needed to improve the learner's mark. Also, the convergence speed is accelerated obviously.

3.3 MTLBO procedure

In order to improve the performance of the conventional TLBO, a sort of new TLBO variant is proposed. Based on the aforementioned explanations of the proposed optimization algorithm, the general procedure of MTLBO algorithm is given in detail as follows.

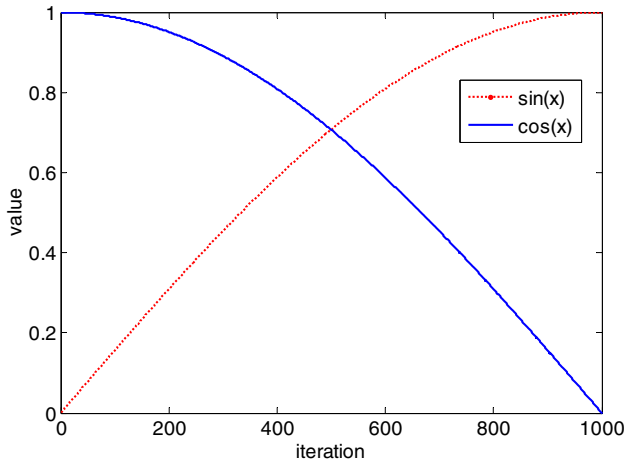

Fig. 1 The simulation curves of $sin(x)$ and $cos(x)$ function

MTLBO Algorithm

1: Initializing algorithm parameters. Population size *Popsize,* Maximum generation *Gen,* Inertia weights ω_{start} , ω_{end} .

2: Random variables population initialization. *gen=0.*

3: while the stopping criteria is not adequate **do**

4 Teaching phase

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

Calculation the mean value X_{mean} of every dimension.

For *i=1 to Popsize* **do**

$T_F = round(1 + rand(0,1))$

$diff = r_i(X_{best} - T_F X_{mean})$

$W = \omega_{start} - (\omega_{start} - \omega_{end}) \times \frac{iter}{MaxIter}$

if $f(X_{old,i}) < f(X_{mean})$

$X_{new,i} = X_{old,i} \times W + (X_{best} - X_{old,i}) \times rand$

else

$X_{new,i} = (X_{old,i} + (rand - 0.5) \times 2 \times (X_{mean} - X_{old,i})) \times \sin(\frac{\pi}{2} \times \frac{iter}{MaxIter}) + diff \times \cos(\frac{\pi}{2} \times \frac{iter}{MaxIter})$

end

Calculate $f(X_{new,i})$

If $f(X_{new,i}) < f(X_{old,i})$

$X_{old,i} = X_{new,i}, f(X_{old,i}) = f(X_{new,i})$

End If

End For

5: Learning phase

Learners are divided into two groups based on fitness values.

For *i=1 to Popsize/2* **do %%For the first group members**

Randomly select a learner $X_{neighbour}$

if $f(X_{old,i}) > f(X_{neighbour})$

$X_{new,i} = X_{old,i} + (X_{neighbour} - X_{old,i}) \times \cos(\frac{\pi}{2} \times \frac{iter}{MaxIter})$

else

$X_{new,i} = X_{old,i} + (rand - 0.5) \times 2 \times (X_{upper\lim it} - X_{lower\lim it})$

end

End For

For *i=1 to Popsize/2* **do %%For the second group members**

$X_{new,i} = X_{old,i} + (X_{best} - X_{old,i}) \times \cos(\frac{\pi}{2} \times \frac{iter}{MaxIter})$

End For

Calculate $f(X_{new})$

If $f(X_{new,i}) < f(X_{old,i})$

$X_{old,i} = X_{new,i}, f(X_{old,i}) = f(X_{new,i})$

End If

gen=gen+1.

End While

Table 1 23 benchmark functions used in experiments

No.	Range	C	Global optima	Formulation
F1	[−100, 100]	US	0	$f(x) = \sum_{i=1}^{n} (x_i)^2$
F2	[−10]	UN	0	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
F3	[−100, 100]	UN	0	$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$
F4	[−100, 100]	UN	0	$f(x) = \max_i \{\lvert x_i \rvert, 1 \leqslant i \leqslant n\}$
F5	[−100, 100]	UN	0	$f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2 - (x_2-\pi)^2)$
F6	[−100, 100]	US	0	$f(x) = \sum_{i=1}^{n} \left(\lfloor x_i + 0.5 \rfloor\right)^2$
F7	[−1.28,1.28]	US	0	$f(x) = \sum_{i=1}^{n} (ix_i)^4 + random[0, 1)$
F8	[−500, 500]	MS	−418.9829×n	$f(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$
F9	[−5.12,5.12]	MS	0	$f(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$
F10	[−600, 600]	MN	0	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F11	[−32, 32]	MN	0	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$
F12	[−50, 50]	MN	0	$f(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})] + (y_n-1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i+1}{4}; \quad u(x_i, a, k, m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i-a)^m & x_i < -a \end{cases}$
F13	[−50, 50]	MN	0	$f(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_i+1)] + (x_n-1)^2[1+\sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$
F14	[−65.53,65.53]	MS	1	$f(x) = \left[\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right]^{-1}$
F15	[−5, 5]	MN	0.00030	$f(x) = \sum_{i=1}^{11}\left(a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right)^2$
F16	[−5, 5]	MN	−1.0316	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
F17	[−5, 10] × [0, 14]	MS	0.398	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$
F18	[−5, 5]	MN	3	$f(x) = [1 + (x_1+x_2+1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$
F19	[0, 1]	MN	−3.86	$f(x) = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2]$
F20	[0, 1]	MN	−3.32	$f(x) = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2]$
F21	[0, 10]	MN	−10.1532	$f(x) = \sum_{i=1}^{5}\sum_{j=1}^{4}[(x_j - a_{ij})(x_j - a_{ij})^T + c_i]^{-1}$
F22	[0, 10]	MN	−10.4028	$f(x) = -\sum_{i=1}^{7}\sum_{j=1}^{4}[(x_j - a_{ij})(x_j - a_{ij})^T + c_i]^{-1}$
F23	[0, 10]	MN	−10.5363	$f(x) = -\sum_{i=1}^{10}\sum_{j=1}^{4}[(x_j - a_{ij})(x_j - a_{ij})^T + c_i]^{-1}$

C characteristics, *U* unimodal, *S* separable, *N* non-separable

4 Experimental study and discussion

4.1 Experimental setup

In this section, to evaluate the performance of MTLBO, 23 famous unconstraint benchmark numerical function problems are adopted, which includes 7 unimodal high dimension functions (F1–F7), 6 multimodal high dimension functions (F8–F13) and 10 fixed dimension functions (F14–F23). For unimodal test functions, every function has only one global optima solution, which makes them be beneficial for verifying the convergence speed and exploitation of

Table 2 Parameters setting

Method	Population size	Maximum iteration	Dimension	others
ABC	40	1000	20, 50	Limit = 200
GSA	40	1000	20, 50	$G_0 = 100$, $\alpha = 20$
SSO	40	1000	20, 50	PF = 0.7
KH	40	1000	20, 50	–
SCA	40	1000	20, 50	a = 2
TLBO	20	1000	20, 50	–
MTLBO	20	1000	20, 50	$\omega_{start} = 0.9$, $\omega_{end} = 0.2$

algorithm. For multimodal test functions, every function has multiple local solutions other than the global optimal solution. These multimodal test functions are suitable to measure the local optima avoidance and the explorative ability of algorithms. These testing functions are described in detail in Table 1, in which includes the searching range, the theory global optima and functional characteristics.

In addition, the performance of the proposed MTLBO algorithm is compared with other heuristic optimization algorithms: artificial bee colony (ABC) [4], Gravitational Search Algorithm (GSA) [5], Krill Herds algorithm (KH) [6], Social-Spider Optimization algorithm (SSO) [7], Sine

Table 3 The mean solution and standard deviation of the 30 trails obtained for 20 dimensional functions

F	Performance	ABC	GSA	SSO	KH	SCA	TLBO	MTLBO
F1	Mean	3.3942×10^{-16}	3.7129×10^{-17}	0.0200	0	1.4334×10^{-9}	4.1139×10^{-217}	**0**
	Std	8.1362×10^{-17}	4.3444×10^{-18}	1.0586×10^{-17}	0	2.4044×10^{-9}	0	**0**
F2	Mean	9.6955×10^{-16}	1.6734×10^{-8}	0.6165	897.1568	5.1521×10^{-9}	1.0161×10^{-123}	$\mathbf{3.9294\times10^{-303}}$
	Std	1.1904×10^{-16}	2.7740×10^{-9}	1.1292×10^{-16}	4.5989×10^{3}	1.1962×10^{-8}	5.2000×10^{-123}	**0**
F3	Mean	2.4931×10^{3}	55.6512	**0.1133**	NaN	105.6656	1.1661×10^{5}	1.4188×10^{5}
	Std	890.0995	31.1602	$\mathbf{8.4690\times10^{-17}}$	NaN	216.7389	5.8675×10^{4}	2.5207×10^{5}
F4	Mean	0.8876	2.7401×10^{-9}	0.0775	29.3327	0.5232	3.7398×10^{-77}	$\mathbf{2.1576\times10^{-295}}$
	Std	0.2650	6.6278×10^{-10}	1.4115×10^{-17}	7.3133	0.9179	2.0149×10^{-76}	**0**
F5	Mean	0.7563	24.9559	81.9765	18.7448	17.9380	18.9149	**0.3947**
	Std	1.3903	28.4303	1.4454×10^{-14}	0.0067	1.3403	0.0796	1.0169
F6	Mean	3.2932×10^{-16}	$\mathbf{1.5688\times10^{-17}}$	0.0383	4.7961	2.0232	3.3132	0.0280
	Std	7.4504×10^{-17}	$\mathbf{5.1923\times10^{-18}}$	0	0.0197	0.2579	0.6164	0.0765
F7	Mean	0.0311	0.0140	0.0238	0.0256	0.0045	0.0033	$\mathbf{5.9881\times10^{-4}}$
	Std	0.0093	0.0059	3.5288×10^{-18}	0.0245	0.0037	0.0020	$\mathbf{5.7362\times10^{-4}}$

Boldface in the tables indicates that the optimization algorithm presents the best performance in several algorithms, which can obtain the theoretical optimal solution or close to the optimal solution of the testing numerical functions

Table 4 The mean solution and standard deviation of the 30 trails obtained for 50 dimensional functions

F	Performance	ABC	GSA	SSO	KH	SCA	TLBO	MTLBO
F1	Mean	6.1565×10^{-16}	1.4246×10^{-16}	0.4024	0	86.5724	6.5515×10^{-212}	**0**
	Std	1.1515×10^{-16}	4.9424×10^{-17}	0	0	272.2453	0	**0**
F2	Mean	2.2326×10^{-5}	0.0648	4.0976	**0**	0.0060	1.3766×10^{-123}	1.1999×10^{-315}
	Std	5.0059×10^{-6}	0.2535	2.7101×10^{-15}	**0**	0.0088	7.3874×10^{-123}	0
F3	Mean	3.7849×10^{4}	1.2271×10^{3}	**76.5628**	NaN	2.8965×10^{4}	6.6681×10^{5}	2.2227×10^{5}
	Std	5.9003×10^{3}	264.7120	$\mathbf{1.4454\times10^{-14}}$	NaN	1.2914×10^{4}	3.7769×10^{5}	3.6225×10^{5}
F4	Mean	36.7053	6.3075	1.2721	36.2370	54.9184	7.0098×10^{-71}	$\mathbf{2.0639\times10^{-301}}$
	Std	2.3312	1.5901	2.2584×10^{-16}	9.7421	8.7282	2.4333×10^{-70}	**0**
F5	Mean	29.9906	73.5339	127.5353	48.5074	3.7847×10^{5}	48.9214	**1.7580**
	Std	32.4871	42.3755	4.3361×10^{-14}	0.0192	7.0952×10^{5}	0.0244	**5.3000**
F6	Mean	4.7381×10^{-9}	$\mathbf{1.6136\times10^{-16}}$	0.4510	12.2656	40.0340	10.8708	0.2016
	Std	4.8158×10^{-9}	$\mathbf{3.7676\times10^{-17}}$	5.6460×10^{-17}	0.0338	51.7003	0.5515	0.4879
F7	Mean	0.2550	0.3216	0.7195	0.0235	0.7172	0.0031	$\mathbf{5.0401\times10^{-4}}$
	Std	0.0496	0.4789	0	0.0174	1.5154	0.0017	$\mathbf{5.7680\times10^{-4}}$

Boldface in the tables indicates that the optimization algorithm presents the best performance in several algorithms, which can obtain the theoretical optimal solution or close to the optimal solution of the testing numerical functions

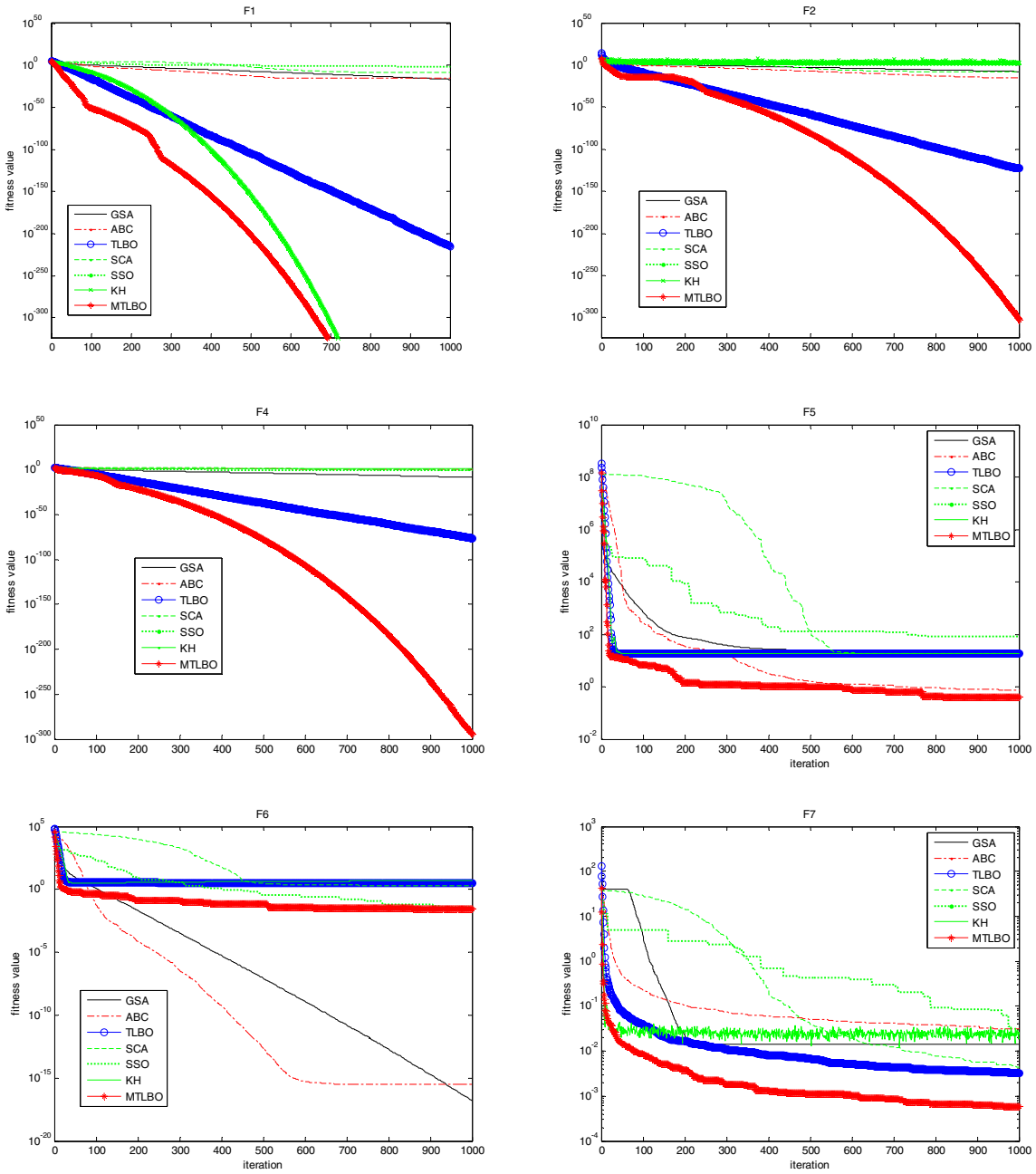

Fig. 2 Convergence curves of the 7 methods on 6 unimodal functions with 20 dimensions

Cosine Algorithm (SCA) [8] and TLBO. They are the state-of-the-art optimization algorithms in recent years. The preset parameters of each algorithm are given in Table 2, where contains population size, maximal iteration and algorithm specific-parameter etc. Shown in Table 2, the maximum iteration indicates that every algorithm stops optimizing after 1000 circles. Dimension represents the dimensions of optimization parameters. For ABC, if a position cannot be improved further through a predetermined number of cycles called Limit then that food source is assumed to be abandoned [4]. For GSA, G_0 is the initial gravitational constant and α is a constant. In [5], the initial gravitational constant G_0 is equal to 100 and α is equal to 20. For SSO, PF is a probability threshold. For SCA, a is a constant, which is set 2 in [8]. Note that: in original TLBO, duplicate elimination process is applied to increase the population diversity. However, the process is not used in our algorithm. So the number of function evaluations of MTLBO algorithm is $=(2\times population\ size\times number\ of\ generations)$. The computation cost will be decreased when the two algorithms have the same

Table 5 The mean solution and standard deviation of the 30 trails obtained for 20 dimensional functions

F	Performance	ABC	GSA	SSO	KH	SCA	TLBO	MTLBO
F8	Mean	-8.3305×10^{3}	-2.3397×10^{3}	-5.5567×10^{3}	-0.0072	-3.2045×10^{3}	-3.7204×10^{3}	$\mathbf{-8.2473\times10^{3}}$
	Std	77.3979	463.0206	0	0.0069	208.8084	459.2280	**584.1095**
F9	Mean	5.3780×10^{-8}	11.3094	34.4531	**0**	2.8124	**0**	**0**
	Std	2.9457×10^{-7}	4.5593	2.1681×10^{-14}	**0**	9.1900	**0**	**0**
F10	Mean	2.3626×10^{-14}	3.7268×10^{-9}	0.1975	$\mathbf{8.8818\times10^{-16}}$	7.0677	5.8620×10^{-15}	$\mathbf{8.8818\times10^{-16}}$
	Std	3.6850×10^{-15}	5.7988×10^{-10}	0	**0**	9.2269	1.7702×10^{-15}	**0**
F11	Mean	2.4863×10^{-4}	1.1012	0.0093	**0**	0.1110	4.9241×10^{-4}	**0**
	Std	0.0013	0.5304	0	**0**	0.2490	0.0027	**0**
F12	Mean	$\mathbf{3.2263\times10^{-16}}$	0.0207	6.0676×10^{-14}	1.8701	0.2869	3.7561	1.3021
	Std	$\mathbf{6.4048\times10^{-17}}$	0.0538	4.4109×10^{-19}	0.0098	0.0596	2.6951	1.4403
F13	Mean	9.6792×10^{-8}	$\mathbf{2.2046\times10^{-32}}$	4.9199×10^{-5}	1.9465	1.1521	0.0896	0.0911
	Std	2.9025×10^{-7}	$\mathbf{2.6147\times10^{-32}}$	0	0.0125	0.1264	0.3019	0.0736

Boldface in the tables indicates that the optimization algorithm presents the best performance in several algorithms, which can obtain the theoretical optimal solution or close to the optimal solution of the testing numerical functions

Table 6 The mean solution and standard deviation of the 30 trails obtained for 50 dimensional functions

F	Performance	ABC	GSA	SSO	KH	SCA	TLBO	MTLBO
F8	Mean	-1.9816×10^{4}	-3.7426×10^{3}	-1.1177×10^{4}	-0.0066	-5.1824×10^{3}	-5.8073×10^{3}	$\mathbf{-2.0811\times10^{4}}$
	Std	270.0780	677.5553	5.5503×10^{-12}	0.0048	333.4967	646.3627	**475.5265**
F9	Mean	1.5958	50.2454	166.8171	**0**	66.0374	**0**	**0**
	Std	1.1231	11.5670	0	**0**	58.4216	**0**	**0**
F10	Mean	1.5896×10^{-4}	7.3771×10^{-9}	1.2179	8.8818×10^{-16}	15.5625	6.3357×10^{-15}	$\mathbf{8.8818\times10^{-16}}$
	Std	6.5017×10^{-4}	1.3053×10^{-9}	2.2584×10^{-16}	0	8.3866	1.8027×10^{-15}	**0**
F11	Mean	0.0029	22.7147	0.0163	0	1.5832	0	**0**
	Std	0.0089	5.5042	0	0	1.1008	0	**0**
F12	Mean	$\mathbf{5.4282\times10^{-11}}$	0.8593	1.4589	1.4572	1.5736×10^{6}	4.1903	1.7068
	Std	$\mathbf{4.0601\times10^{-11}}$	0.5038	2.2584×10^{-16}	0.0036	3.1857×10^{6}	2.8568	1.7145
F13	Mean	$\mathbf{6.4083\times10^{-7}}$	0.1561	3.7347×10^{-4}	4.8963	2.9214×10^{6}	1.5130	2.1563
	Std	$\mathbf{1.3216\times10^{-6}}$	0.3800	2.7568×10^{-19}	0.0257	3.9542×10^{6}	7.1825	1.9684

Boldface in the tables indicates that the optimization algorithm presents the best performance in several algorithms, which can obtain the theoretical optimal solution or close to the optimal solution of the testing numerical functions

maximum generations. In this paper, the population size of the TLBO and MTLBO algorithm is set to 20.

The tests are implemented on Intel (R) Core(TM)64 × 2 Dual Core Processor T5670 @1.80, 1.79 GHz and 2 GB RAM. All algorithms are coded and carried out in Matlab 2009 version under the Windows XP Professional.

4.2 Comparison with other algorithms

In this subsection, simulations using MTLBO, ABC, GSA, KH, SSO, SCA and TLBO are conducted on all the above 23 functions. Due to the stochastic nature of meta-heuristics, the results of one single run may be unreliable. Therefore, each algorithm runs 30 times independently to reduce the statistical error. The performance of different optimization algorithms in terms of the mean and standard deviation (SD) of solutions obtained in the 30 independent runs for 20 and 50 dimensional functions. The maximal iteration 1000 is used as the stopping criterion.

4.2.1 The analysis of experimental results for unimodal functions

In this subsection, 7 unimodal functions are adopted to evaluate the performance of MTLBO. The experimental results of 20 and 50 dimensional functions are listed in Tables 3 and 4, respectively. Boldface in the tables indicates the best results. The mean value is smaller, the performance of algorithm is better. The standard deviation value is lower, the stability of algorithm is stronger. For these tables, it is

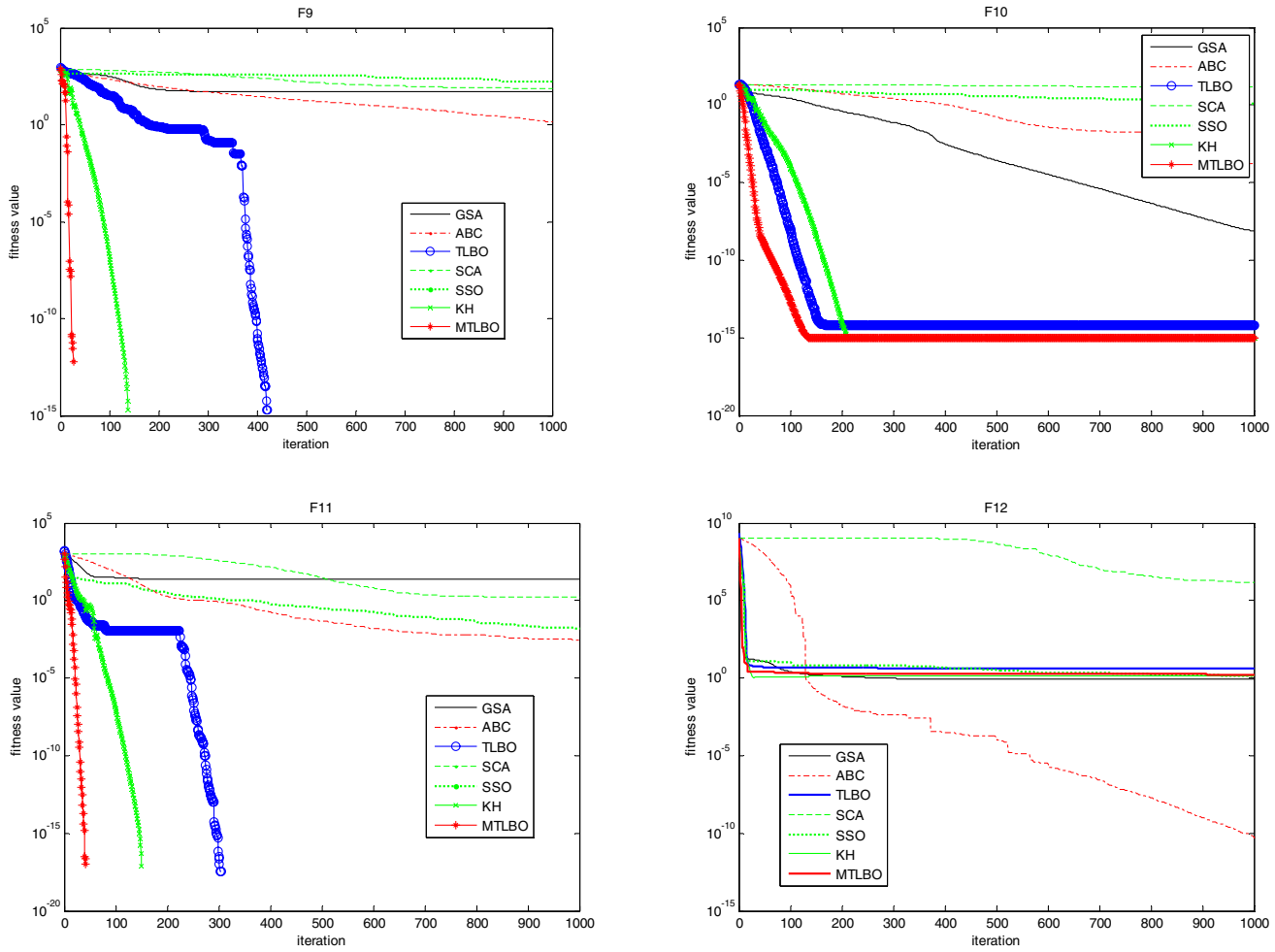

Fig. 3 Convergence curves of the 7 methods on 4 multimodal functions with 50 dimensions

easy to see that MTLBO algorithm wins the smallest mean value and standard deviation value in 5 unimodal functions. Additionally, Fig. 2 graphically presents the comparison in terms of convergence speed and solution quality for solving 6 unimodal functions (F1, F2, F4, F5, F6 and F7) with 20 dimensions. The detailed analysis of experimental results is given as follows.

Tables 3 and 4 show that MTLBO outperforms some other methods in terms of the mean and standard deviation for some unimodal functions. Shown in Table 3, the performance of MTLBO is the best than all other methods for functions F1, F2, F4, F5 and F7. The GSA has the smallest mean and standard deviation for function F6. The SSO displays the best performance for function F3. Particularly, compared with TLBO, the MTLBO shows better performance for functions F1, F2, F4, F5, F6 and F7. The results reveal that MTLBO algorithm enhances the solution quality. Table 4 indicates that the mean and the standard deviation of MTLBO is the best than all other algorithms for functions F1, F4, F5 and F7. The GSA also has the smallest mean and standard deviation for function F6. The SSO still displays the best performance for function F3. The KH shows the best performance for functions F1 and F2. Compared with TLBO, MTLBO shows better performance for all the 7 functions. In short, MTLBO outperforms all the other algorithms on most functions.

4.2.2 The analysis of experimental results for multimodal functions

In this subsection, 6 multimodal functions are used to evaluate the performance of MTLBO. The experimental results of 20 and 50 dimensional functions are listed in Tables 5 and 6, respectively. Boldface in the tables indicates the best results. The mean value is smaller, the performance of algorithm is better. The standard deviation value is lower, the stability of algorithm is stronger. According to these tables, the proposed MTLBO algorithm presents superior performance

Table 7 The comparison results of fixed dimension functions for 7 optimization algorithms

Function	ABC		GSA		SSO		KH		SCA		TLBO		MTLBO	
	Runtime	Mean	Runtime	Mean	Runtime	Mean	Runtime	Mean	Runtime	Mean	Runtime	Mean	Runtime	Mean
F14	4.8568	0.9980	5.0220	0.9980	8.0570	3.9683	5.8876	1.0019	2.1245	0.9980	2.2857	0.9980	2.2342	0.9980
F15	1.6942	3.2168e−4	3.0434	0.0017	3.8349	3.6970e−4	3.1687	3.0774e−4	0.3070	3.2791e−4	0.6467	3.3697e−4	0.4233	5.7014e−4
F16	1.6130	− 1.0316	2.8093	− 1.0316	3.6390	− 1.0316	3.0691	− 0.0195	0.2241	− 1.0316	0.4772	− 1.0316	0.3556	− 0.8442
F17	1.5242	0.3979	2.8063	0.3979	3.5556	0.3979	3.0357	0.4072	0.1995	0.3986	0.4608	0.3979	0.3408	0.3981
F18	1.4565	3.0000	2.8118	3.0000	3.5849	3.0001	3.0580	0.4261	0.2081	3.0000	0.4630	3.0000	0.3402	3.3628
F19	1.8671	− 3.8628	3.1018	− 3.4196	3.6191	− 3.8627	3.1748	− 0.1710	0.4104	− 3.8560	0.6629	− 3.8626	0.5062	− 3.4541
F20	1.8838	− 3.3219	3.2624	− 1.8949	3.5480	− 3.1970	3.2236	− 0.0130	0.4311	− 2.8023	0.8679	− 3.1243	0.5309	− 2.1032
F21	2.2476	− 10.1532	3.2233	− 4.8370	3.8430	− 10.1444	3.3050	− 0.2905	0.5708	− 2.5511	0.8395	− 6.4420	0.6233	− 10.1532
F22	2.2127	− 10.4029	3.3123	− 7.2138	4.0469	− 10.3967	3.4384	− 0.3112	0.6764	− 4.3569	0.9494	− 7.2937	0.7277	− 10.4028
F23	2.3570	− 10.5364	3.4724	− 10.1759	4.2396	− 10.5319	3.5731	− 0.3395	0.8225	− 4.8630	1.0988	− 7.6780	0.8879	10.5363

on most functions. Additionally, Fig. 3 graphically presents the comparison in terms of convergence speed and solution quality for solving 4 unimodal functions (F9, F10, F11 and F12) with 50 dimensions. The detailed analysis of experimental results is given as follows.

Seen from Table 5, the performance of MTLBO is better than other algorithms for functions F8, F9, F10 and F11. The ABC has the smallest mean and standard deviation for function F12. The GSA has better performance in terms of the mean and the standard deviation than all other methods for function F13. The KH shows the best performance than all other algorithms for functions F9, F10 and F11. The original TLBO has good performance for function F9. Table 6 shows the performance results for 50 dimensional functions. For this table, it is easy to observe that MTLBO wins the smallest mean and standard deviation for functions F8, F9, F10 and F11 as well. Moreover, the ABC has the smallest mean and standard deviation in two functions F12 and F13. The KH algorithm also obtains the better performance for functions F9, F10 and F11. Compared with TLBO, MTLBO shows better performance for all functions except function F13. In brief, the proposed MTLBO improves the solution quality for multi-modal functions.

4.2.3 The analysis of experimental results for fixed dimension functions

In this subsection, 10 fixed dimension functions are applied to evaluate the performance of the proposed modified teaching–learning-based optimization algorithm. The experimental results are listed in Table 7. According to this Table, the ABC can achieve the global optimal solution on all functions. The GSA can find the global optima on 4 functions (F14, F16, F17, F18). The SSO algorithm can find the global optima or the solutions close to the global optima on 9 functions except F14. The KH can find the global optima only on two functions (F14, F15). The SCA can find the global solutions on 6 functions (F14, F15, F16, F17, F18, F19). For TLBO, it can achieve the global optima solution on 7 functions (F14, F15, F16, F17, F18, F19, F20). The MTLBO can find the global optima on six functions (F14, F17, F18, F21, F22, F23) with short running time. For the remaining functions, MTLBO can obtain the solution quite close to the global optima. Particularly, compared with TLBO, the running time of MTLBO is shorter on all fixed dimension functions. It is noted that although the solution updating mechanism of MTLBO is more complex than the original TLBO, the MTLBO algorithm eliminates the duplicate elimination process. Hence, the computation cost will be decreased when the two methods have the same maximum generation. Moreover, three inertia weights are introduced in MTLBO to enhance the convergence speed and solution

Table 8 The mean solution and standard deviation of the 30 trails obtained for 10 dimensional functions

Function	TLBO		MTLBO	
	Mean	Std	Mean	Std
Shifted and rotated bent cigar function	2.1789×10^{10}	7.5819×10^{9}	2.0798×10^{10}	7.7529×10^{9}
Shifted and rotated zakharov function	1.6606×10^{6}	6.9452×10^{6}	1.2711×10^{6}	2.6771×10^{6}
Shifted and rotated rosenbrock's function	3.4615×10^{3}	1.3754×10^{3}	3.0728×10^{3}	1.4009×10^{3}
Shifted and rotated rastrigin's function	659.3708	25.0948	665.6474	23.1643
Shifted and rotated expanded scaffer's F6 function	700.5771	14.2159	700.0188	16.8315
Shifted and rotated lunacek bi_rastrigin function	1.1650×10^{3}	89.8692	1.2042×10^{3}	107.6970
Shifted and rotated non-continuous rastrigin's function	941.3247	19.3184	940.2373	19.8744
Shifted and rotated levy function	5.5234×10^{3}	1.5285×10^{3}	6.0253×10^{3}	1.6432×10^{3}
Shifted and rotated schwefel's function	3.4838×10^{3}	268.7526	3.7045×10^{3}	356.4339
Hybrid function 1 (N = 3)	2.6565×10^{4}	2.9365×10^{4}	2.7699×10^{4}	2.4414×10^{4}
Hybrid function 2(N = 3)	2.7798×10^{9}	1.4411×10^{9}	2.9408×10^{9}	1.5693×10^{9}
Hybrid function 3 (N = 3)	2.5563×10^{8}	2.4573×10^{8}	2.8851×10^{8}	3.6181×10^{8}
Hybrid function 4 (N = 4)	8.2390×10^{6}	1.1475×10^{7}	1.1691×10^{7}	2.2147×10^{7}
Hybrid function 5 (N = 4)	3.0534×10^{7}	3.8550×10^{7}	1.6513×10^{7}	3.7513×10^{7}
Hybrid function 6 (N = 4)	2.8254×10^{3}	208.9106	2.8057×10^{3}	229.4395
Hybrid function 6 (N = 5)	2.5095×10^{3}	228.6194	2.4121×10^{3}	228.4598
Hybrid function 6 (N = 5)	8.9181×10^{8}	7.7793×10^{8}	9.8545×10^{8}	7.5464×10^{8}
Hybrid function 6 (N = 5)	9.1999×10^{7}	1.2794×10^{8}	1.1065×10^{8}	1.8413×10^{8}
Hybrid function 6 (N = 6)	2.5313×10^{3}	114.5715	2.5755×10^{3}	136.1357
Composition function 1 (N = 3)	2.4257×10^{3}	59.9058	2.4487×10^{3}	28.6270
Composition function 2 (N = 3)	4.1366×10^{3}	426.7760	4.2685×10^{3}	613.4919
Composition function 3 (N = 4)	2.8117×10^{3}	37.1548	2.8441×10^{3}	50.7518
Composition function 4 (N = 4)	3.0234×10^{3}	89.2576	3.0302×10^{3}	102.9255
Composition function 5 (N = 5)	4.7568×10^{3}	820.9488	4.6981×10^{3}	879.5401
Composition function 6 (N = 5)	4.9994×10^{3}	430.6828	5.0198×10^{3}	515.2874
Composition function 7 (N = 6)	3.4271×10^{3}	126.5702	3.4305×10^{3}	100.7306
Composition function 8 (N = 6)	4.1905×10^{3}	219.3734	4.1905×10^{3}	230.7724
Composition function 9 (N = 3)	3.9582×10^{3}	201.7234	3.9729×10^{3}	184.4599

quality. Therefore, MTLBO is able to obtain good solution with low simulation time.

4.3 Comparisons with TLBO on CEC2017 benchmark functions

In this subsection, the performance of MTLBO is compared with TLBO on 28 CEC2017 benchmark functions, which are obtained from the literature [28]. These functions contain 2 unimodal functions, 7 simple multimodal functions, 10 hybrid functions and 9 composition functions. The detailed descriptions of these functions are recorded in literature [28]. For both MTLBO and TLBO, the population size is 40 and the maximum iteration is 500. In order to reduce the computation error, MTLBO and TLBO separately run 30 times for every testing function. Average results of 30 trials of simulations are listed in Tables 8, 9 and 10, respectively.

Seen from Tables 8, 9 and 10, TLBO and MTLBO gain similar performance on most testing functions. However, the two algorithms cannot get the global optima for every testing function. Therefore, TLBO and MTLBO are suitable to solve the basic benchmark functions that have global optima in zero point. In the future, we will focus on solving the problem of TLBO and MTLBO.

5 Conclusions

In this paper, a variant of TLBO which is called MTLBO is proposed to enhance the solution quality and convergence speed of TLBO. The proposed MTLBO algorithm firstly

Table 9 The mean solution and standard deviation of the 30 trails obtained for 30 dimensional functions

Function	TLBO		MTLBO	
	Mean	Std	Mean	Std
Shifted and rotated Bent cigar function	1.2284×10^{11}	2.0771×10^{10}	1.2737×10^{11}	1.8538×10^{10}
Shifted and rotated zakharov function	4.6740×10^{9}	2.0136×10^{10}	5.2896×10^{9}	1.8405×10^{10}
Shifted and rotated rosenbrock's function	5.5946×10^{4}	1.3124×10^{4}	5.4755×10^{4}	1.3595×10^{4}
Shifted and rotated rastrigin's function	1.1844×10^{3}	60.5419	1.2014×10^{3}	55.1426
Shifted and rotated expanded scaffer's F6 function	741.6269	11.7887	737.8008	11.7779
Shifted and rotated lunacek bi_rastrigin function	3.5356×10^{3}	317.4131	3.4276×10^{3}	262.7589
Shifted and rotated non-continuous rastrigin's function	1.4159×10^{3}	53.2604	1.4146×10^{3}	54.5449
Shifted and rotated levy function	4.1115×10^{4}	6.1019×10^{3}	3.7494×10^{4}	7.2148×10^{3}
Shifted and rotated schwefel's function	9.9218×10^{3}	488.0740	1.0917×10^{4}	486.5566
Hybrid function 1 (N = 3)	1.4090×10^{5}	2.8593×10^{5}	1.1163×10^{6}	4.5772×10^{6}
Hybrid function 2(N = 3)	2.9353×10^{10}	6.5555×10^{9}	2.6977×10^{10}	8.0056×10^{9}
Hybrid function 3 (N = 3)	2.8415×10^{10}	7.5450×10^{9}	2.9294×10^{10}	8.5243×10^{9}
Hybrid function 4 (N = 4)	7.1860×10^{7}	5.2320×10^{7}	8.9172×10^{7}	6.2885×10^{7}
Hybrid function 5 (N = 4)	7.8007×10^{9}	2.9895×10^{9}	8.3547×10^{9}	3.1943×10^{9}
Hybrid function 6 (N = 4)	9.3028×10^{3}	2.1722×10^{3}	9.5515×10^{3}	2.7252×10^{3}
Hybrid function 6 (N = 5)	7.6930×10^{4}	1.1215×10^{5}	3.5421×10^{4}	3.6445×10^{4}
Hybrid function 6 (N = 5)	8.2693×10^{8}	5.7650×10^{8}	9.0937×10^{8}	6.1028×10^{8}
Hybrid function 6 (N = 5)	9.2381×10^{9}	4.1533×10^{9}	9.2560×10^{9}	3.5055×10^{9}
Hybrid function 6 (N = 6)	3.9225×10^{3}	257.5821	3.9576×10^{3}	144.7786
Composition function 1 (N = 3)	2.9323×10^{3}	59.2757	2.9414×10^{3}	53.1039
Composition function 2 (N = 3)	1.1560×10^{4}	614.2084	1.2360×10^{4}	517.3640
Composition function 3 (N = 4)	3.9044×10^{3}	178.3639	4.0075×10^{3}	234.8605
Composition function 4 (N = 4)	4.0858×10^{3}	192.2295	4.3857×10^{3}	224.2766
Composition function 5 (N = 5)	1.9252×10^{4}	3.7711×10^{3}	1.7118×10^{4}	3.7649×10^{3}
Composition function 6 (N = 5)	1.6058×10^{4}	1.7721×10^{3}	1.5894×10^{4}	1.7017×10^{3}
Composition function 7 (N = 6)	5.5456×10^{3}	809.0739	5.6290×10^{3}	555.7922
Composition function 8 (N = 6)	1.3723×10^{4}	2.0055×10^{3}	1.3862×10^{4}	1.9271×10^{3}
Composition function 9 (N = 3)	7.7406×10^{4}	9.7483×10^{4}	1.2417×10^{5}	2.0214×10^{5}

introduces the concept of 'actual teaching–learning situation' so that the population updating mechanism is conformed to the real class. Also, three inertia weights are used to enhance the solution quality and convergence speed and balance the exploration and exploitation of MTLBO. The performance of MTLBO algorithm is verified over a series of 23 unconstraint benchmark numerical function problems. The results reveal that the proposed MTLBO outperforms other state-of-the-art algorithms on most functions. In addition, 28 CEC2017 benchmark functions are used to verify the performance of MTLBO. The experimental results reveal that the current version of MTLBO can not present satisfactory performance on these testing functions.

In the future, the MTLBO will be improved and used to solve more challenging rotated and shifted functions and optimize the structure of artificial neural network. Simultaneously, it will be used for real life problems.

Table 10 The mean solution and standard deviation of the 30 trails obtained for 50 dimensional functions

Function	TLBO		MTLBO	
	Mean	Std	Mean	Std
Shifted and rotated Bent cigar function	2.5903×10^{11}	2.8852×10^{10}	2.6674×10^{11}	2.1434×10^{10}
Shifted and rotated zakharov function	2.5655×10^{9}	9.6719×10^{9}	6.2570×10^{10}	1.4337×10^{11}
Shifted and rotated rosenbrock's function	1.1126×10^{5}	2.0436×10^{4}	1.0797×10^{5}	2.3177×10^{4}
Shifted and rotated rastrigin's function	1.7198×10^{3}	70.3659	1.7083×10^{3}	57.6134
Shifted and rotated expanded scaffer's F6 function	757.0666	9.1033	755.1756	10.1274
Shifted and rotated lunacek bi_rastrigin function	6.1337×10^{3}	419.4560	6.0469×10^{3}	362.5893
Shifted and rotated non-continuous rastrigin's function	2.0390×10^{3}	68.2537	1.9997×10^{3}	57.6396
Shifted and rotated levy function	1.0345×10^{3}	1.7060×10^{4}	1.0595×10^{5}	1.3770×10^{4}
Shifted and rotated schwefel's function	1.6780×10^{4}	1.0590×10^{3}	1.8304×10^{4}	596.6899
Hybrid function 1 (N = 3)	2.5248×10^{6}	1.2790×10^{7}	1.0781×10^{6}	2.1249×10^{6}
Hybrid function 2(N = 3)	1.4392×10^{11}	2.6339×10^{10}	1.4337×10^{11}	2.4483×10^{10}
Hybrid function 3 (N = 3)	8.8356×10^{10}	2.2191×10^{10}	8.6713×10^{10}	1.9413×10^{10}
Hybrid function 4 (N = 4)	3.8686×10^{8}	1.8054×10^{8}	5.8389×10^{8}	2.6144×10^{8}
Hybrid function 5 (N = 4)	3.6048×10^{10}	9.4571×10^{9}	3.1757×10^{10}	1.1224×10^{10}
Hybrid function 6 (N = 4)	1.6472×10^{4}	3.0691×10^{3}	1.6093×10^{4}	3.8245×10^{3}
Hybrid function 6 (N = 5)	1.9735×10^{6}	2.4855×10^{6}	1.9927×10^{6}	2.4767×10^{6}
Hybrid function 6 (N = 5)	1.6670×10^{9}	7.9373×10^{8}	1.6564×10^{9}	8.9225×10^{8}
Hybrid function 6 (N = 5)	1.5270×10^{10}	4.2319×10^{9}	1.3497×10^{10}	3.5258×10^{9}
Hybrid function 6 (N = 6)	5.5450×10^{3}	273.6023	5.5895×10^{3}	272.0417
Composition function 1 (N = 3)	3.5900×10^{3}	73.0091	3.6064×10^{3}	59.7891
Composition function 2 (N = 3)	1.9172×10^{4}	978.9190	1.9977×10^{4}	596.9000
Composition function 3 (N = 4)	5.0062×10^{3}	171.7641	5.2637×10^{3}	287.5217
Composition function 4 (N = 4)	5.4382×10^{3}	370.7366	5.8402×10^{3}	388.5443
Composition function 5 (N = 5)	6.2025×10^{4}	7.8288×10^{3}	6.0369×10^{4}	1.2059×10^{4}
Composition function 6 (N = 5)	3.2652×10^{4}	3.6587×10^{3}	3.2039×10^{4}	3.3209×10^{3}
Composition function 7 (N = 6)	7.9665×10^{3}	1.7975×10^{3}	9.2417×10^{3}	1.0412×10^{3}
Composition function 8 (N = 6)	2.5413×10^{4}	2.6396×10^{3}	2.4900×10^{4}	4.3670×10^{3}
Composition function 9 (N = 3)	2.3745×10^{6}	2.5631×10^{6}	2.4251×10^{6}	2.1769×10^{6}

Acknowledgements The authors make a grateful acknowledgement for the associate editor and the reviewers for their valuable time and thoughtful comments to the improvement of the paper. This work is supported by the National Natural Science Foundation of China (Grant no. 61573306).

References

1. Rao RV, Savsani VJ, Vakharia DP (2012) Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 183(1):1–15
2. Doğan B, Ölmez T (2015) A new metaheuristic for numerical function optimization: vortex search algorithm. Inf Sci 293:125–145
3. Kennedy J. Particle swarm optimization[M]//encyclopedia of machine learning. Springer US, 2010: 760–766
4. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471
5. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
6. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845
7. Cuevas E, Cienfuegos M, Zaldívar D et al (2013) A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Syst Appl 40(16):6374–6384
8. Mirjalili SSCA. (2016) A Sine Cosine Algorithm for solving optimization problems. Knowl Based Syst 96:120–133
9. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
10. Li G, Niu P, Xiao X (2012) Development and investigation of efficient artificial bee colony algorithm for numerical function optimization. Appl Soft Comput 12(1):320–332
11. Mernik M, Liu SH, Karaboga D et al (2015) On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation. Inf Sci 291:115–127
12. Petrović M, Vuković N, Mitić M et al (2016) Integration of process planning and scheduling using chaotic particle swarm optimization algorithm. Expert Syst Appl 64:569–588
13. Mandal B, Roy PK, Mandal S (2014) Economic load dispatch using krill herd algorithm. Int J Electr Power Energy Syst 57:1–10

14. Niu P, Ma Y, Li M et al (2016) A kind of parameters self-adjusting extreme learning machine. Neural Process Lett 44(3):813–830
15. Ghasemi M, Ghavidel S, Rahmani S et al (2014) A novel hybrid algorithm of imperialist competitive algorithm and teaching learning algorithm for optimal power flow problem with non-smooth cost functions. Eng Appl Artif Intell 29:54–69
16. Niknam T, Fard AK, Baziar A (2012) Multi-objective stochastic distribution feeder reconfiguration problem considering hydrogen and thermal energy production by fuel cell power plants. Energy 42(1):563–573
17. Niknam T, Azizipanah-Abarghooee R, Narimani MR (2012) An efficient scenario-based stochastic programming framework for multi-objective optimal micro-grid operation. Appl Energy 99:455–470
18. Niknam T, Azizipanah-Abarghooee R, Narimani MR (2012) A new multi objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems. Eng Appl Artif Intell 25(8):1577–1588
19. Niknam T, Golestaneh F, Sadeghi MS (2012) θ-Multiobjective teaching–learning-based optimization for dynamic economic emission dispatch. Syst J IEEE 6(2):341–352
20. Rao RV, Kalyankar VD (2013) Parameter optimization of modern machining processes using teaching–learning-based optimization algorithm. Eng Appl Artif Intell 26(1):524–531
21. Rao RV, Kalyankar VD (2012) Multi-objective multi-parameter optimization of the industrial LBW process using a new optimization algorithm. Proc Inst Mech Eng Part B J Eng Manuf 226(6):1018–1025
22. Krishnanand KR, Panigrahi BK, Rout PK et al (2011) Application of multi-objective teaching–learning-based algorithm to an economic load dispatch problem with incommensurable objectives[M]//swarm, evolutionary, and memetic computing. Springer, Berlin, pp 697–705
23. Li G, Niu P, Zhang W et al (2013) Model NO*x* emissions by least squares support vector machine with tuning based on ameliorated teaching–learning-based optimization. Chemometr Intell Lab Syst 126:11–20
24. Rao R, Patel V (2013) Comparative performance of an elitist teaching–learning-based optimization algorithm for solving unconstrained optimization problems. Int J Ind Eng Comput 4(1):29–50
25. Rao RV, Patel V (2013) An improved teaching–learning-based optimization algorithm for solving unconstrained optimization problems. Scientia Iranica 20(3):710–720
26. Roy PK, Bhui S (2013) Multi-objective quasi-oppositional teaching learning based optimization for economic emission load dispatch problem. Int J Electr Power Energy Syst 53:937–948
27. Črepinšek M, Liu SH, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput Surv 45(3):35
28. Awad NH, Ali MZ, Liang JJ, Qu BY, Suganthan PN (2016) Problem definitions and evaluation criteria for the CEC2017 special session and competition on single objective bound constrained real-parameter numerical optimization, Technical Report, Nanyang Technological University, Singapore