


# Fuzzy adaptive teaching–learning-based optimization for global numerical optimization

Min-Yuan Cheng<sup>1</sup> · Doddy Prayogo<sup>1,2</sup> 

Received: 25 November 2015 / Accepted: 1 July 2016 / Published online: 14 July 2016  
© The Natural Computing Applications Forum 2016

**Abstract** Teaching–learning-based optimization (TLBO) is one of the latest metaheuristic algorithms being used to solve global optimization problems over continuous search space. Researchers have proposed few variants of TLBO to improve the performance of the basic TLBO algorithm. This paper presents a new variant of TLBO called fuzzy adaptive teaching–learning-based optimization (FATLBO) for numerical global optimization. We propose three new modifications to the basic scheme of TLBO in order to improve its searching capability. These modifications consist, namely of a status monitor, fuzzy adaptive teaching–learning strategies, and a remedial operator. The performance of FATLBO is investigated on four experimental sets comprising complex benchmark functions in various dimensions and compared with well-known optimization methods. Based on the results, we conclude that FATLBO is able to deliver excellence and competitive performance for global optimization.

**Keywords** Metaheuristic · Global optimization · Fuzzy logic · Teaching–learning-based optimization · Parameter tuning

## 1 Introduction

The field of metaheuristics has emerged as a promising research area in which those involved focus on solving various optimization problems by using the simulation of a variety of natural phenomena. Genetic algorithm (GA) [1], particle swarm optimization (PSO) [2], and differential evolution (DE) [3] are a few examples of some early metaheuristic algorithms that have been proposed throughout the past decades. Metaheuristic algorithms have now been used in many research areas, including those of computer science [4], forecasting [5, 6], medicine [7], economics [8], and engineering [9, 10]. The success of those metaheuristic algorithms in solving various optimization problems across a wide range of fields has resulted in proposals for promising algorithms like artificial bee colony (ABC) [11], multi-verse optimizer (MVO) [12], symbiotic organisms search (SOS) [13], and Jaya algorithm [14]. One of the advantages of metaheuristic algorithms is that they do not need to deeply adapt to each problem in order to solve it, which is contrary to problem-specific heuristics [15]. According to the “No-Free-Lunch” theorem [16], any pair of search algorithms has similar average performance with regard to all possible optimization problems. However, this does not exclude the possibility that certain algorithms will obtain better results for some objective functions. Thus, the need to develop high-performance metaheuristic algorithms increases continually.

Inspired by the idea of the classroom teaching–learning process, Rao et al. developed a metaheuristic algorithm entitled the teaching–learning-based optimization (TLBO) algorithm to solve various constrained [17] and unconstrained benchmark test functions [18]. TLBO mimics the natural phenomena of the classroom.

---

✉ Doddy Prayogo  
doddyprayogo@gmail.com; prayogo@petra.ac.id  
Min-Yuan Cheng  
myc@mail.ntust.edu.tw

<sup>1</sup> Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei 106, Taiwan

<sup>2</sup> Department of Civil Engineering, Petra Christian University, Jalan Siwalankerto 121-131, Surabaya 60236, Indonesia

For the TLBO algorithm, the term “student” is used to represent a candidate solution to the corresponding optimization problem. Meanwhile, the term “teacher” is deemed to be the best learner in the classroom, and it is used to represent the current best solution. All students in the classroom individually perform unique search patterns over solution space through the teacher phase and the learner phase. The teacher phase is characterized by the transmission of knowledge from a teacher to students. The learner phase, on the other hand, is characterized by the transmission of knowledge from mutual interactions between students. Metaheuristic algorithms in general require extra algorithm-specific control parameters. Thus, the advantage the TLBO has over majority of metaheuristic algorithms is that it only requires control parameters such as maximum number of generations and population size.

Since it was initially proposed, there have been numerous applications of the TLBO in solving different complex problems. Some are utilized in data clustering [19], mechanical engineering [17], structural engineering [20], parameter optimization [21], and economic dispatch [22]. Several discussions related to TLBO have also appeared in the literature surrounding this topic [23, 24]. As it is a new and promising optimization algorithm, researchers worldwide have attempted to enhance the performance of TLBO. Rao and Patel [25] added the elitism concept, where the elite solutions replace the worst solutions. To make TLBO faster and more robust, Satapathy and Naik [26] introduced a modified TLBO (*m*TLBO), which altered the learner phase formula by adopting extra training through tutorials. Rao and Patel [27] presented several new elements in the improved TLBO (I-TLBO) such as adaptive teaching factors, number of teachers, tutorial training, and self-motivated learning to increase the searching ability of TLBO. Previous research findings have successfully improved the capability of the basic TLBO, and thus, developing better metaheuristic algorithms remains a top priority as the amount of hard and complex problems have continued to augment.

The fact that the TLBO algorithm is relatively new yet demonstrates superior performance over other metaheuristic algorithms in solving various benchmarks and engineering problems, has drawn our attention and encouraged us to further investigate and enhance the capability of the TLBO algorithm. In this paper, a new metaheuristic algorithm called fuzzy adaptive teaching–learning-based optimization (FATLBO) is introduced to improve the performance of its predecessor. Three new modifications are implemented: a status monitor, fuzzy adaptive teaching–learning strategies (FATLS), and a remedial operator. These modifications aim to increase the

effectiveness and efficiency of TLBO. In the status monitor, a new procedure to track the productivity of each student in both the teacher and learner phases is developed. FATLS uses a fuzzy logic controller to control the operation of teacher and learner phases, setting a much larger weight on the most productive phase and thus increasing the convergence speed. A remedial operator is deployed to activate the restart mechanism for the entire classroom if there is no significant improvement over a certain period of time. We compare FATLBO with a number of algorithms, including the basic TLBO, ITLBO, and other well-known metaheuristic algorithms across a range of benchmark functions. Experimental results confirm the excellent performance of proposed FATLBO algorithm in comparison with the other metaheuristic algorithms.

The remaining sections are organized as follows. In Sect. 2, we describe the regular TLBO algorithm. In Sect. 3, we briefly review previous works related to parameter adaptation techniques. In Sect. 4, we describe the proposed algorithm in further detail. In Sect. 5, we present the experimental results. Finally, Sect. 6 consists of our conclusions.

## 2 The teaching–learning-based optimization (TLBO) algorithm

Rao et al. [17, 18] introduced TLBO as a population-based metaheuristic algorithm that simulates traditional classroom learning. First, a random population consisting of students is generated based on population size and the number of design variables. A student,  $X_i$ , is represented by one solution vector to a specific objective function with a corresponding fitness value,  $f(X_i)$ . The searching procedure is performed throughout two main phases of the learning process: the teacher phase and the learner phase. The efficiency of knowledge transfer is determined by the appropriate combination of these two phases. In the teacher phase, a teacher is required to deliver knowledge directly to students. After evaluating the fitness value of each student, the teacher ( $X_{\text{teacher}}$ ) role is assigned to the smartest student with the lowest fitness value in that iteration. The other students are then improved by shifting the average knowledge of all students ( $X_{\text{mean}}$ ) toward the  $X_{\text{teacher}}$  as shown in Eq. (1):

$$X_{\text{new}} = X_i + \text{rand} \times (X_{\text{teacher}} - T_F \times X_{\text{mean}}), \quad (1)$$

where  $\text{rand}$  is a uniform random number between 0 and 1 and  $T_F$  is a teaching factor with a random value of either 1 or 2.

In the learner phase, a student ( $X_i$ ) engages with other students and benefits from the exchange, which is selected randomly ( $X_j$ ). The student  $X_i$  is first compared with student

$X_j$ . If student  $X_j$  is smarter than  $X_i$ , then  $X_i$  is shifted toward  $X_j$  as shown in Eq. (2). Otherwise, it is shifted away from  $X_j$  as shown in Eq. (3):

$$X_{i\text{new}} = X_i + \text{rand} \times (X_j - X_i), \tag{2}$$

$$X_{i\text{new}} = X_i + \text{rand} \times (X_i - X_j). \tag{3}$$

For both the teacher and learner phases, the fitness value of each student is allowed to update only when the new solution ( $X_{i\text{new}}$ ) outperforms the former one ( $X_i$ ). Subsequently, the TLBO attempts to enhance some students by modifying them throughout the teacher and learner phases. The algorithm is run until the maximum number of iterations is reached. The whole main procedure of the TLBO algorithm is shown in Fig. 1.

### 3 Previous works related to parameter adaptation techniques

It is worth noting that although TLBO is described in the literature as an algorithm-specific parameter-less optimization method, common control parameters (such as population size and maximum number of iterations) still need to be determined. Nevertheless, unlike most meta-heuristic algorithms, TLBO does not have any algorithm-specific control parameter. Hence, no additional effort is required to set additional parameters for the algorithm to work. This situation offers a huge benefit to TLBO since choosing the proper control parameters requires some expertise. The inappropriate choice of control parameters may contribute to worse performance.

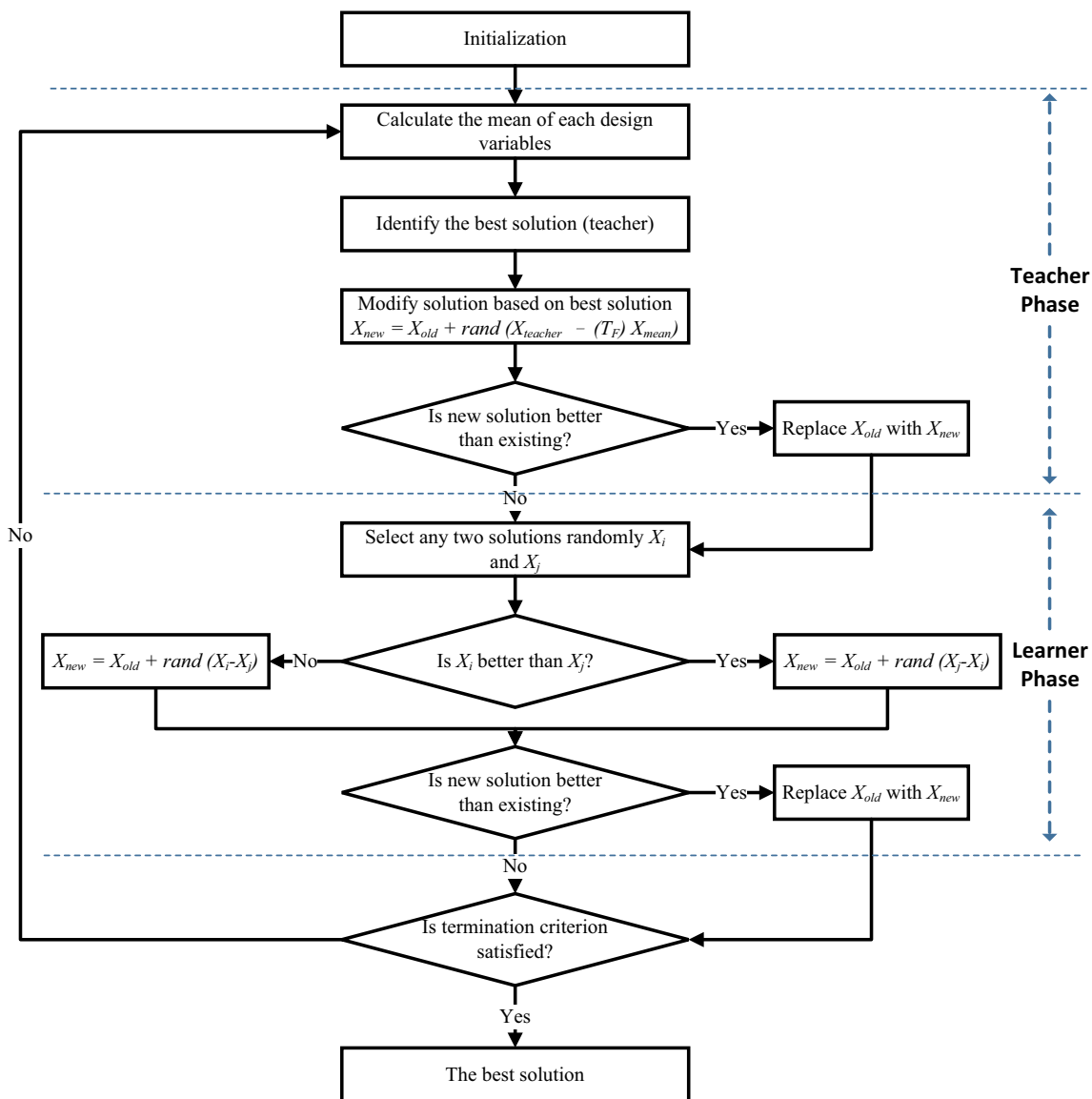


Fig. 1 TLBO flowchart

Despite these above-mentioned advantages, the lack of additional algorithm-specific parameters also entails some drawbacks. The additional algorithm-specific parameters help metaheuristic algorithms generate some unique searching patterns to solve various optimization problems comprising a variety of fitness landscapes [28]. In GA, after the selection process, the solutions undergo crossover and mutation operators with certain probabilities. The crossover and mutation operators occur only within some probabilities, called the crossover rate (CR) and the mutation rate (MR), respectively. When the solutions are not subjected to crossover or mutation, they remain unmodified. CR and MR are categorized as algorithm-specific control parameters in GA. The choice of CR and MR is known to significantly affect the behavior and performance of GA [29]. Similarly to GA, TLBO has two operators that modify the current solutions, known as teacher phase and learner phase. However, unlike GA, these two operators are not subjected to probabilities. The solutions will undergo teacher and learner phases throughout the iterations. With respect to CR and MR, adding algorithm-specific parameters that control the probabilities of both the teacher and learner phases may increase the performance of TLBO.

Many researchers have developed parameter adaptation techniques for finding the optimal parameter setting instead of setting fixed values to the control parameters. For example, Qin et al. [30] proposed a self-adaptive DE algorithm (SaDE) that employs mutation strategies and the respective control parameter is self-adapted based on their previous experiences of generating promising solutions. Yang and Hsieh [31] developed auto-tuning boundary-approaching PSO (AB-PSO) algorithm, which has the feature of adaptive auto-tuning parameters. The adjustment is conducted after a certain number of iterations, allowing some time for the control parameters to make an impact.

Meanwhile, some researchers have experimented with using fuzzy concepts to improve the performance of different types of optimization algorithms. Shi and Eberhart [32] proposed fuzzy adaptive PSO to improve the performance of the PSO by dynamically adjusting the inertia weight parameter by integrating a fuzzy system with fitness feedback of each particle. Melin et al. [33] improved the swarm diversity and convergence capability of PSO using fuzzy parameter adaptation for designing the fuzzy classifier. Several schemes of Mamdani's fuzzy inference systems were developed to adjust the cognitive factor ( $c_1$ ) and social factor ( $c_2$ ) parameters using the information of iteration, swarm diversity, and swarm convergence. Using fuzzy logic controllers whose inputs incorporate the relative function values, Liu and Lampinen [34] proposed fuzzy adaptive DE (FADE) to make some parameter

adjustments for the mutation and crossover operations. The obtained results show that the FADE algorithm has a better performance than a regular DE in regard to higher-dimensional problems.

In summary, the setting of algorithm-specific parameters with the appropriate adaptive strategy may affect the optimization performance of metaheuristic algorithms significantly. Introducing some new algorithm-specific control parameters and implementing the right strategy for its parameter adaptation may improve the performance of TLBO.

#### 4 Fuzzy adaptive teaching–learning-based optimization (FATLBO)

In the real classroom, teaching methods are paramount for the successful transmission of teachers' knowledge to students. Certain materials are adapted to the teacher-centered approach, whereas the others are adequate for the so-called student-centered approach. The term student-centered approach refers to the teaching style in which students are involved in discussions and learning through cooperation. It is important to underline that inadequate teaching methods are likely to have an adverse effect on the learning process.

In the fundamental form of the TLBO algorithm, students are required to experience both teacher and learner phases. Accordingly, the TLBO needs two function evaluations to calculate the fitness value of one candidate solution during one iteration. This paper presents the implementation of the teaching methods used in the real classroom through the usage of modifications of the fundamental TLBO. Section 4.1 will elaborate on the status monitor we developed. This monitor records the progress of students related to both phases. It has the capacity to measure the productivity in both phases, as well as the improvement of the knowledge of students in a given time frame of an iteration. Section 4.2 presents the new strategy called the fuzzy adaptive teaching–learning system (FATLS), to choose appropriate teaching methods in order to optimize the efficiency and the search speed of the TLBO algorithm. We allow the students to skip the teacher or learner phase by introducing two new parameters, called the teaching rate (TR) and the learning rate (LR). TR and LR present the probabilities that a student will enter the teacher and learner phases. If TR is set to 0.7, then every student will have a 70 % probability of moving to the teacher phase and 30 % probability of skipping the teacher phase. Fuzzy logic is further employed to self-adjust the TR and LR based on information collected from the status monitor. Finally, in Sect. 4.3, we introduce a remedial operator to ensure students are not trapped in local optima.

### 4.1 Status monitor

Two new variables are introduced to measure the productivity of students during the teacher and learner phases: the success rate of teacher phase (SRTP) and the success rate of learner phase (SRLP). The values of SRTP and SRLP vary between 0 and 1, where a higher value means the more success in improving student knowledge.

Table 1 explains the impact of the SRTP and the SRLP calculation during one iteration. As can be seen from Table 1, the value of the SRTP is 0.4, whereas the value of the SRLP is 0.6. These results show that entering the teacher phase is going to enhance the fitness for about 40 % of students and entering the learner phase is going to enhance the fitness for about 60 % of students. The advantage of this monitor is that it follows the progress of students in all iterations. FATLS uses the SRTP and the SRLP information obtained by the status monitor to adapt the fittest TR and LR for each student in the following step.

### 4.2 Fuzzy adaptive teaching–learning strategies (FATLS)

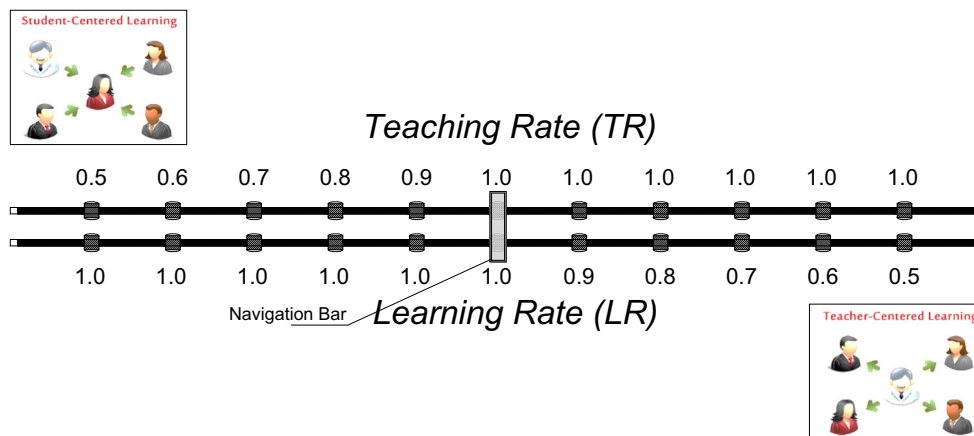
The fuzzy adaptive system has been developed to boost productivity during the learning process. The SRTP and SRLP information gathered by the status monitor is further used as input variables. In addition, regarding output variables, we proposed a bar controller to adapt the LR and the TR. Figure 2 represents an illustration of this bar controller. The central bar is actually the navigation bar. When the bar goes toward the left, the TR is in decline. It practically means that the FATLS will focus on the learner phase. On the contrary, when the bar goes toward the right, the LR will be in decline. *Bar movement* is used as the output variable. Then, we adjust the bar movement to tune the TR and LR.

The FATLS consists of three principal components: fuzzification, inferencing process, and defuzzification.

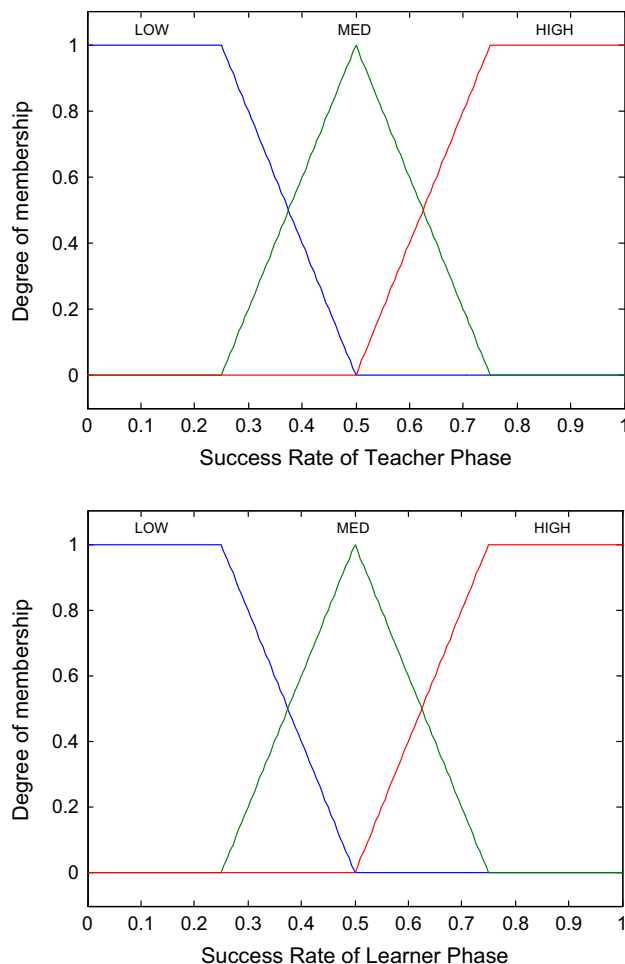
**Table 1** Example of success rate calculation

Individual	Fitness value from the end of last iteration	Fitness value after teacher phase	Success?	Fitness value after learner phase	Success?
1	5.49	<b>3.29</b>	Yes	3.29	No
2	19.97	<b>13.1</b>	Yes	13.1	No
3	68.69	68.69	No	68.69	No
4	61.95	<b>35.18</b>	Yes	<b>28.79</b>	Yes
5	74.89	74.89	No	<b>50.78</b>	Yes
6	54.11	54.11	No	<b>47.13</b>	Yes
7	22.32	22.32	No	<b>17.66</b>	Yes
8	48.51	<b>47.56</b>	Yes	<b>46.11</b>	Yes
9	3.04	3.04	No	3.04	No
10	6.68	6.68	No	<b>6.06</b>	Yes
Success rate			4/10		6/10

The number in bold italics indicates the updated fitness value



**Fig. 2** Fuzzy bar controller



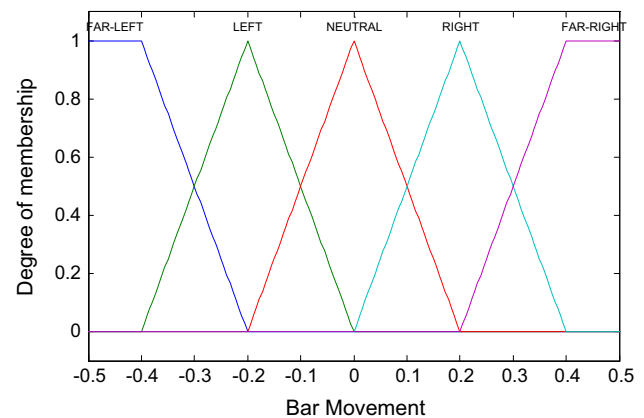
**Fig. 3** Membership function of the SRTP and SRLP

#### 4.2.1 Fuzzification

Figure 3 shows the conversion of all input variables from the preceding step into the membership grade. Left-triangle, triangle, and right-triangle functions are used as the membership functions for all input variables. Figure 4 shows five triangular membership functions used for the output variable, namely one left-triangle, three triangles, and one right-triangle.

#### 4.2.2 Inferencing process

The Mamdani-type fuzzy rules are employed to formulate the conditional statements that consist of the fuzzy rule base. The Mamdani's fuzzy inference system is also applied to map all the input variables to the target output. As previously mentioned, the input variables are success rate of teacher phase (SRTP) and success rate of learner phase (SRLP), while the output variable is bar movement. The bar movement is critical to the tuning of the teaching



**Fig. 4** Membership function of the bar movement

rate (TR) and learning rate (LR) parameters. The proposed mapping process by the fuzzy inference system is illustrated in Fig. 5. Meanwhile, the following nine fuzzy rules aim to develop the best parameter adaptation scheme for the TR and LR parameters:

1. **If (SRTP is LOW) and (SRLP is LOW), then (bar movement is NEUTRAL).**
2. **If (SRTP is LOW) and (SRLP is MEDIUM), then (bar movement is LEFT).**
3. **If (SRTP is LOW) and (SRLP is HIGH), then (bar movement is FAR-LEFT).**
4. **If (SRTP is MEDIUM) and (SRLP is LOW), then (bar movement is RIGHT).**
5. **If (SRTP is MEDIUM) and (SRLP is MEDIUM), then (bar movement is NEUTRAL).**
6. **If (SRTP is MEDIUM) and (SRLP is HIGH), then (bar movement is LEFT).**
7. **If (SRTP is HIGH) and (SRLP is LOW), then (bar movement is FAR-RIGHT).**
8. **If (SRTP is HIGH) and (SRLP is MEDIUM), then (bar movement is RIGHT).**
9. **If (SRTP is HIGH) and (SRLP is HIGH), then (bar movement is NEUTRAL).**

All the fuzzy rules are designed to choose the teaching method that will have the most impact on the learning process. The bar controller plays a key role in controlling the values of TR and LR. If the bar moves to the left, then it lowers the TR and maintains the high rate of LR. This forces the students to attend the learner phase more frequently than the teacher phase, thereby mimicking a student-centered approach. Conversely, if the bar moves to the right, then it lowers the LR and maintains the high rate of TR. This forces students to attend the teacher phase more frequently than the learner phase, thereby creating a teaching-centered approach. In the inferencing process, the bar movement will adapt through different combinations of

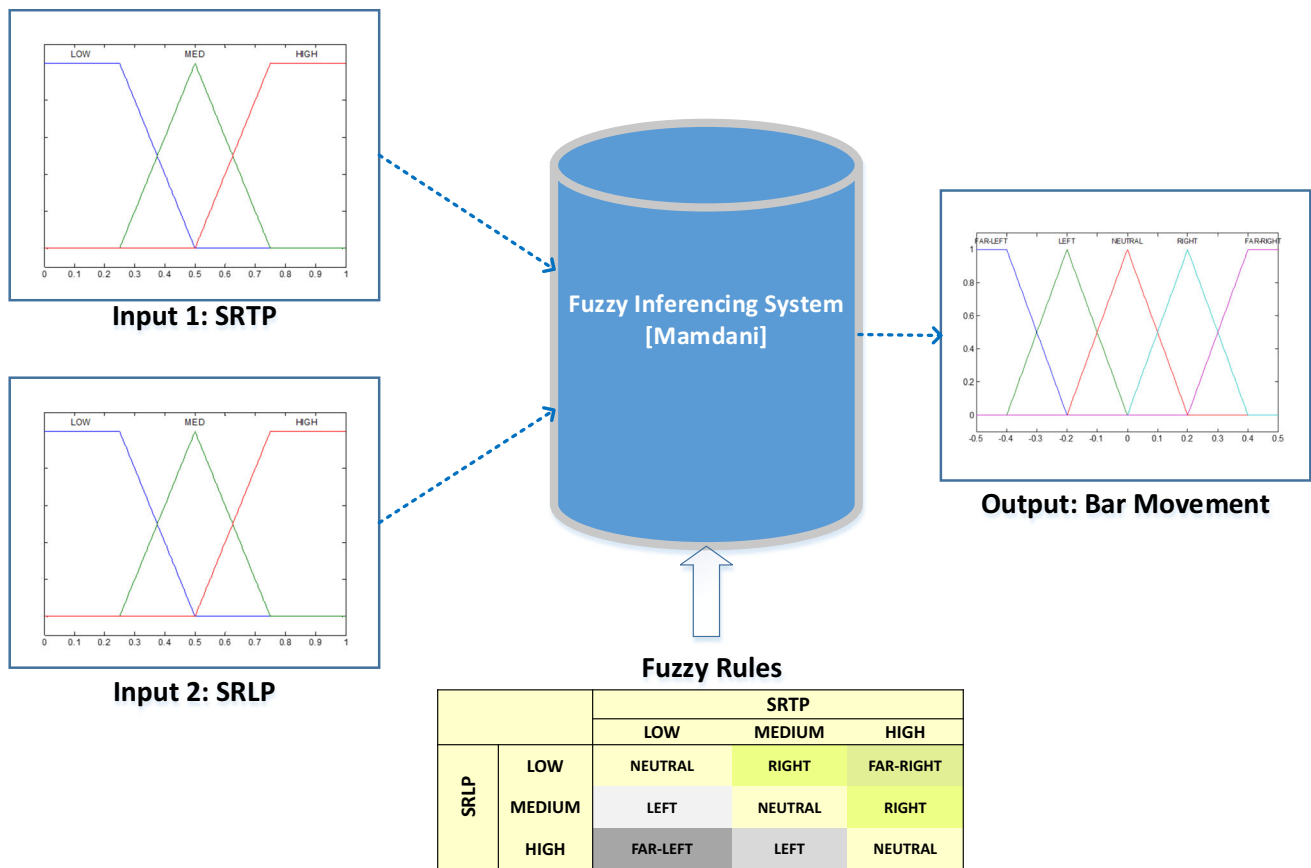


Fig. 5 The mapping process by Mamdani’s fuzzy inference system

SRTP and SRLP, which were previously calculated from the status monitor. When SRTP is high and SRLP is low, it indicates that students’ knowledge is improving mainly due to the effect of the teacher phase rather than that of the learner phase. Therefore, it is reasonable for FATLBO to switch the current teaching–learning strategy into a more teacher-centered approach by moving the bar controller to the right (decreasing LR while maintaining TR) and vice versa.

4.2.3 Defuzzification

The next part of the process is the opposite of fuzzification. After the process of inferencing is complete, defuzzification is begun. Center-of-sums is the method we chose for defuzzification. Fuzzy strategies have an output, named bar movement, and based on that, the LR and TR of the following iterations can be adjusted. As an example, if there is a bar movement result of  $-0.35$ , the bar is going to travel 3.5 steps toward the left, which means that the TR and LR will be adjusted to 0.65 and 1.0, respectively, as shown in Fig. 6.

4.3 Remedial operator

There are times in class when whole groups of students show no improvement in the process of learning. This could be due to several factors, such as bad environment, difficult topic, or teaching style. Faced with this issue, the teacher could evaluate it and as a result add remediation or perturbation to the classroom based on the current circumstances. Remediation such as this may include downgrading the content and/or volume of the lesson, the application of personal tutoring, or a change in the style of teaching.

On the occurrence of premature convergence, the remedial operator adjusts the environment of the classroom by a population expansion utilizing the random step based on the Gaussian law, with a center point of the teacher. In this paper, we assume premature convergence if fitness value does not improve after a long period or if the whole population makes a convergence in a single solution.

The formula for the remedial operator is shown in Eq. (4).

$$X_{new} = X_{teacher} + N(0, range/2), \tag{4}$$

where range is equal to upper and lower bounds.

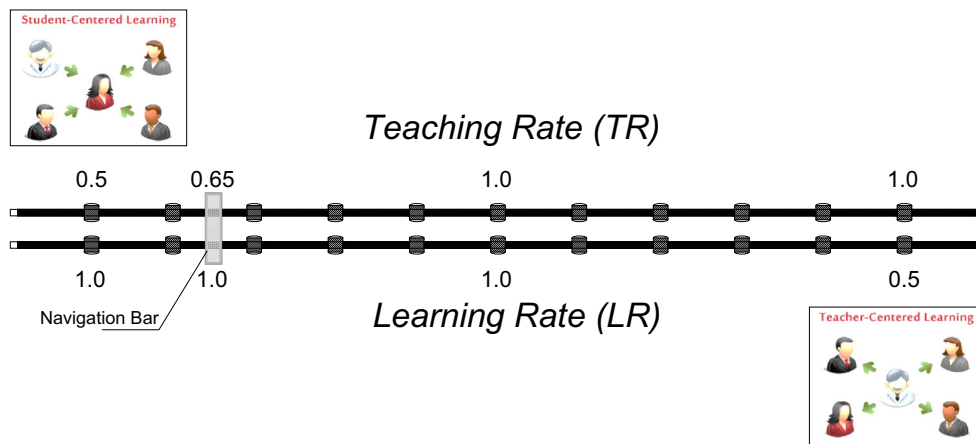


Fig. 6 Fuzzy bar controller after FATLS applied

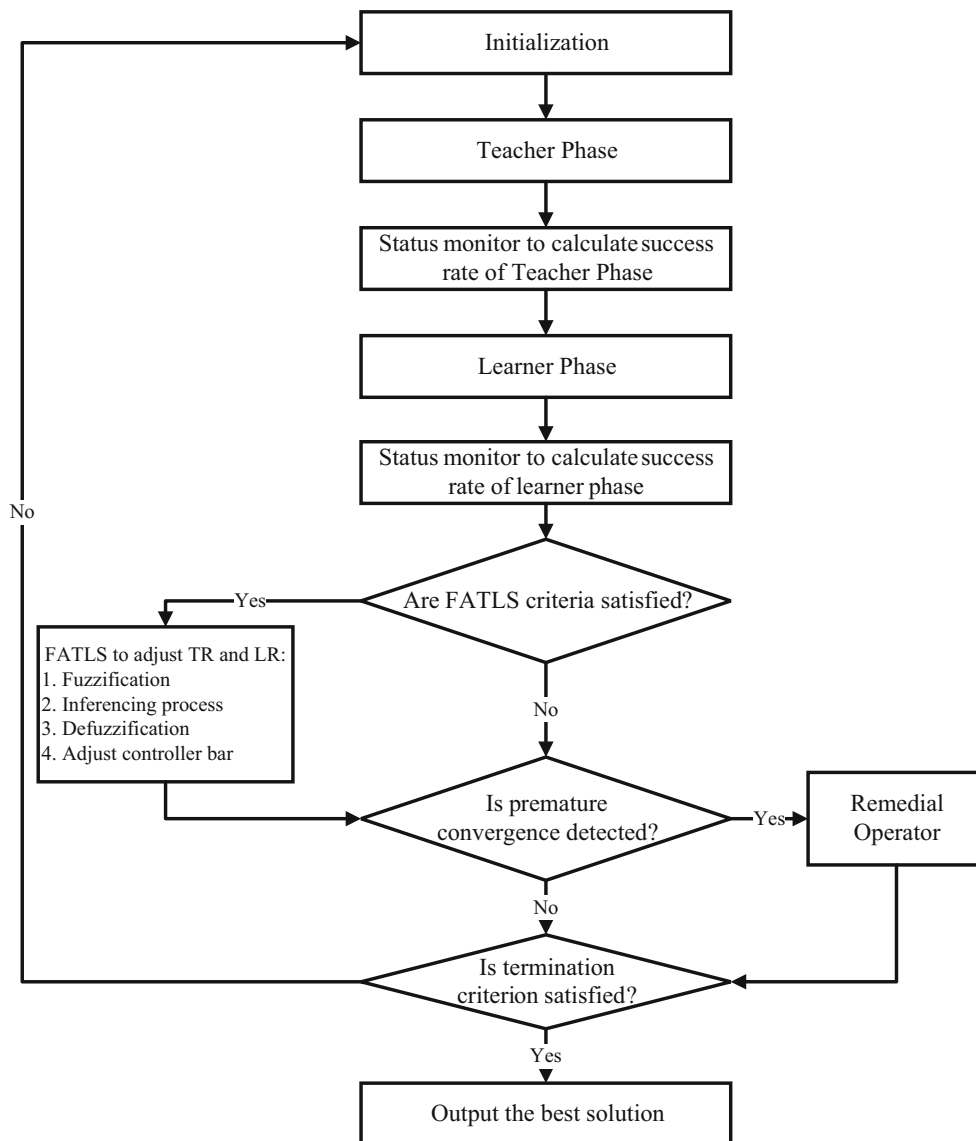


Fig. 7 FATLBO flowchart



To conclude, the entire procedure of FATLBO is shown in Fig. 7.

### 5 Experimental results

We compare the FATLBO performance with other meta-heuristic algorithms from the literature, including TLBO, ABC, PSO, DE, and their variants. This section is typically divided into four subsections that explain the four different experiment sets, including the details regarding benchmark functions, the setup of the experiment, and the results.

#### 5.1 Experiment 1

In this experiment, we apply FATLBO to eight different benchmark functions that were originally solved by Liang et al. [35]. Table 2 gives the details of the benchmark functions. Previously, all functions were tested with the maximum of 30,000 function evaluations. To maintain consistency in the comparison, the FATLBO algorithm is also tested with the same maximum number of function evaluations. For the population size, FATLBO uses 10 individuals for all functions except Function 2, which uses 25 individuals.

Each benchmark function is tested 30 times with the FATLBO algorithm, and results are obtained in the form of mean value and standard deviation. FATLBO is compared with TLBO variants [27], ABC variants [36], and PSO variants [35]. With the exception of FATLBO, the rest of the results are obtained from those previous works. The comparative results of FATLBO and other algorithms are shown in Table 3.

The FATLBO algorithm identified better solutions compared with other algorithms in 6 functions: Sphere, Rosenbrock, Griewank, Weierstrass, Rastrigin, and NCRastrigin functions. Meanwhile, I-TLBO, TLBO, CLPSO, and modified ABC have only found 5, 5, 4, and 1 solution(s) better than others, respectively. FATLBO is able to achieve the global optimum in the Sphere, Griewank, Weierstrass, Rastrigin, and NCRastrigin functions. Furthermore, the FATLBO algorithm performs better than the rest of the considered algorithms for the Rosenbrock function. Although FATLBO was not able to find a solution better than other algorithms in the Ackley and Schwefel functions, FATLBO can consistently produce a strong performance and is still better than more than half of its competitors.

Although the results presented in Table 3 provide first insight into the performance of the algorithms, carrying out a statistical hypothesis test is still required for a better comparison. For this purpose, an unpaired *t* test is performed with a statistical significance value  $\alpha = 0.05$ .

**Table 2** Details of benchmark functions in Experiment 1 (*D*: dimensions)

No.	Function	Formulation	<i>D</i>	Search range	Initialization range
1	Sphere	$F_{\min} = \sum_{i=1}^D x_i^2$	10	[- 100, 100]	[- 100, 50]
2	Rosenbrock	$F_{\min} = \sum_{i=1}^D [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$	10	[- 2.048, 2.048]	[- 2.048, 2.048]
3	Ackley	$F_{\min} = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i \right) + 20 + e$	10	[- 32.768, 32.768]	[- 32.768, 16]
4	Griewank	$F_{\min} = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$	10	[- 600, 600]	[- 600, 200]
5	Weierstrass	$F_{\min} = \sum_{i=1}^D \left[ \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k(x_i + 0.5))] \right] - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)]$ , $a = 0.5, b = 3, k_{\max} = 20$	10	[- 0.5, 0.5]	[- 0.5, 0.2]
6	Rastrigin	$F_{\min} = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	10	[- 5.12, 5.12]	[- 5.12, 2]
7	NCRastrigin	$F_{\min} = \sum_{i=1}^D [y_i^2 - 10 \cos(2\pi y_i) + 10]$ , $y_i = \begin{cases} x_i, &  x_i  < 0.5, \\ \text{round}(2x_i), &  x_i  \geq 0.5 \end{cases}$	10	[- 5.12, 5.12]	[- 5.12, 2]
8	Schwefel	$F_{\min} = - \sum_{i=1}^D (x_i \sin(\sqrt{ x_i }))$	10	[- 500, 500]	[- 500, 500]

**Table 3** Comparative results of TLBO and other algorithms in Experiment 1

	Mean $\pm$ SD Sphere	Mean $\pm$ SD Rosenbrock	Mean $\pm$ SD Ackley	Mean $\pm$ SD Griewank
PSO-w	7.96E-051 $\pm$ 3.56E-050	3.08E+000 $\pm$ 7.69E-001	1.58E-014 $\pm$ 1.60E-014	9.69E-002 $\pm$ 5.01E-002
PSO-cf	9.84E-105 $\pm$ 4.21E-104	6.98E-001 $\pm$ 1.46E+000	9.18E-001 $\pm$ 1.01E+000	1.19E-001 $\pm$ 7.11E-002
PSO-w-local	2.13E-035 $\pm$ 6.17E-035	3.92E+000 $\pm$ 1.19E+000	6.04E-015 $\pm$ 1.67E-015	7.80E-002 $\pm$ 3.79E-002
PSO-cf-local	1.37E-079 $\pm$ 5.60E-079	8.60E-001 $\pm$ 1.56E+000	5.78E-002 $\pm$ 2.58E-001	2.80E-002 $\pm$ 6.34E-002
UPSO	9.84E-118 $\pm$ 3.56E-117	1.40E+000 $\pm$ 1.88E+000	1.33E+000 $\pm$ 1.48E+000	1.04E-001 $\pm$ 7.10E-002
FDR	2.21E-090 $\pm$ 9.88E-090	8.67E-001 $\pm$ 1.63E+000	3.18E-014 $\pm$ 6.40E-014	9.24E-002 $\pm$ 5.61E-002
FIPS	3.15E-030 $\pm$ 4.56E-030	2.78E+000 $\pm$ 2.26E-001	3.75E-015 $\pm$ 2.13E-014	1.31E-001 $\pm$ 9.32E-002
CPSO-H	4.98E-045 $\pm$ 1.00E-044	1.53E+000 $\pm$ 1.70E+000	1.49E-014 $\pm$ 6.97E-015	4.07E-002 $\pm$ 2.80E-002
CLPSO	5.15E-029 $\pm$ 2.16E-28	2.46E+000 $\pm$ 1.70E+000	4.32E-014 $\pm$ 2.55E-014	4.56E-003 $\pm$ 4.81E-003
ABC	7.09E-017 $\pm$ 4.11E-017	2.08E+000 $\pm$ 2.44E+000	4.58E-016 $\pm$ 1.76E-016	1.57E-002 $\pm$ 9.06E-003
Modified ABC	7.04E-017 $\pm$ 4.55E-017	4.42E-001 $\pm$ 8.67E-001	<i>3.32E-016 <math>\pm</math> 1.84E-016</i>	1.52E-002 $\pm$ 1.28E-002
TLBO	<i>0.00 <math>\pm</math> 0.00</i>	1.72E+00 $\pm$ 6.62E-01	3.55E-15 $\pm$ 8.32E-31	<i>0.00 <math>\pm</math> 0.00</i>
I-TLBO	<i>0.00 <math>\pm</math> 0.00</i>	2.00E-01 $\pm$ 1.42E-01	1.42E-15 $\pm$ 1.83E-15	<i>0.00 <math>\pm</math> 0.00</i>
FATLBO	<i>0.00 <math>\pm</math> 0.00</i>	<i>1.55E-04 <math>\pm</math> 2.67E-04</i>	3.97E-15 $\pm$ 1.23E-15	<i>0.00 <math>\pm</math> 0.00</i>
	Weierstrass	Rastrigin	NCRastrigin	Schwefel
PSO-w	2.28E-003 $\pm$ 7.04E-003	5.82E+000 $\pm$ 2.96E+000	4.05E+000 $\pm$ 2.58E+000	3.20E+002 $\pm$ 1.85E+002
PSO-cf	6.69E-001 $\pm$ 7.17E-001	1.25E+001 $\pm$ 5.17E+000	1.20E+001 $\pm$ 4.99E+000	9.87E+002 $\pm$ 2.76E+002
PSO-w-local	1.41E-006 $\pm$ 6.31E-006	3.88E+000 $\pm$ 2.30E+000	4.77E+000 $\pm$ 2.84E+000	3.26E+002 $\pm$ 1.32E+002
PSO-cf-local	7.85E-002 $\pm$ 5.16E-002	9.05E+000 $\pm$ 3.48E+000	5.95E+000 $\pm$ 2.60E+000	8.78E+002 $\pm$ 2.93E+002
UPSO	1.14E+000 $\pm$ 1.17E+00	1.17E+001 $\pm$ 6.11E+000	5.85E+000 $\pm$ 3.15E+000	1.08E+003 $\pm$ 2.68E+002
FDR	3.01E-003 $\pm$ 7.20E-003	7.51E+000 $\pm$ 3.05E+000	3.35E+000 $\pm$ 2.01E+000	8.51E+002 $\pm$ 2.76E+002
FIPS	2.02E-003 $\pm$ 6.40E-003	2.12E+000 $\pm$ 1.33E+000	4.35E+000 $\pm$ 2.80E+000	7.10E+001 $\pm$ 1.50E+002
CPSO-H	1.07E-015 $\pm$ 1.67E-015	<i>0.00 <math>\pm</math> 0.00</i>	2.00E-001 $\pm$ 4.10E-001	2.13E+002 $\pm$ 1.41E+002
CLPSO	<i>0.00 <math>\pm</math> 0.00</i>	<i>0.00 <math>\pm</math> 0.00</i>	<i>0.00 <math>\pm</math> 0.00</i>	<i>0.00 <math>\pm</math> 0.00</i>
ABC	9.01E-006 $\pm$ 4.61E-005	1.61E-016 $\pm$ 5.20E-016	6.64E-017 $\pm$ 3.96E-017	7.91E+000 $\pm$ 2.95E+001
Modified ABC	<i>0.00 <math>\pm</math> 0.00</i>	1.14E-007 $\pm$ 6.16E-007	1.58E-011 $\pm$ 7.62E-011	3.96E+000 $\pm$ 2.13E+001
TLBO	2.42E-05 $\pm$ 1.38E-20	6.77E-08 $\pm$ 3.68E-07	2.65E-08 $\pm$ 1.23E-07	2.94E+02 $\pm$ 2.68E+02
I-TLBO	<i>0.00 <math>\pm</math> 0.00</i>	<i>0.00 <math>\pm</math> 0.00</i>	<i>0.00 <math>\pm</math> 0.00</i>	1.10E+02 $\pm$ 1.06E+02
FATLBO	<i>0.00 <math>\pm</math> 0.00</i>	<i>0.00 <math>\pm</math> 0.00</i>	<i>0.00 <math>\pm</math> 0.00</i>	1.72E-02 $\pm$ 1.06E+02

The number in italics indicates the best value

Table 4 shows the statistical results of FATLBO compared with those of other algorithms. The last column of Table 4 presents the total count of the three statistically significant conclusions of performance comparisons between FATLBO and other algorithms from the  $t$  test (a/b/c). According to the table, FATLBO outperforms the rest of the algorithms, including the PSO and variants, the original TLBO, and the improved version of TLBO (I-TLBO). In comparison with FATLBO, the second best algorithm is I-TLBO, where FATLBO has a significantly better performance in two functions, a statistically similar performance in five functions, and a significantly worse performance in only one function.

## 5.2 Experiment 2

In this experiment, we apply FATLBO to eight different benchmark functions that Mathur et al. [37] previously solved. The benchmark function set is solved later by Pham et al. [38], Ahrari and Atai [39], and Rao and Patel [27]. The results obtained using the FATLBO algorithm are compared with the previous works of TLBO variants along with other well-known optimization algorithms. The details of the benchmark functions are shown in Table 5.

We maintain consistency in the comparison between previous works by adhering to the same rules. In previous studies, the optimization process stops when the difference

**Table 4** Statistical comparison of FATLBO with other optimization algorithms using an unpaired *t* test in Experiment 1

Algorithms	<i>t</i> value with $\alpha = 0.05$ and degree of freedom = 58								Total (a/b/c)
	Sphere	Rosenbrock	Ackley	Griewank	Weierstrass	Rastrigin	NCRastrigin	Schwefel	
PSO-w	1.225 <sup>b</sup>	21.936 <sup>a</sup>	4.038 <sup>a</sup>	10.594 <sup>a</sup>	1.774 <sup>a</sup>	10.769 <sup>a</sup>	8.598 <sup>a</sup>	8.220 <sup>a</sup>	7/1/0
PSO-cf	1.280 <sup>b</sup>	2.618 <sup>a</sup>	4.978 <sup>a</sup>	9.167 <sup>a</sup>	5.111 <sup>a</sup>	13.243 <sup>a</sup>	13.172 <sup>a</sup>	18.285 <sup>a</sup>	7/1/0
PSO-w-local	1.890 <sup>a</sup>	18.042 <sup>a</sup>	5.467 <sup>a</sup>	11.272 <sup>a</sup>	1.224 <sup>b</sup>	9.240 <sup>a</sup>	9.199 <sup>a</sup>	10.547 <sup>a</sup>	7/1/0
PSO-cf-local	1.340 <sup>b</sup>	3.019 <sup>a</sup>	1.227 <sup>b</sup>	2.419 <sup>a</sup>	8.333 <sup>a</sup>	14.244 <sup>a</sup>	12.534 <sup>a</sup>	15.434 <sup>a</sup>	6/2/0
UPSO	1.514 <sup>b</sup>	4.078 <sup>a</sup>	4.922 <sup>a</sup>	8.023 <sup>a</sup>	5.337 <sup>a</sup>	10.488 <sup>a</sup>	10.172 <sup>a</sup>	20.525 <sup>a</sup>	7/1/0
FDR	1.225 <sup>b</sup>	2.913 <sup>a</sup>	2.381 <sup>a</sup>	9.021 <sup>a</sup>	2.290 <sup>a</sup>	13.487 <sup>a</sup>	9.129 <sup>a</sup>	15.765 <sup>a</sup>	7/1/0
FIPS	3.784 <sup>a</sup>	67.371 <sup>a</sup>	-0.057 <sup>b</sup>	7.699 <sup>a</sup>	1.729 <sup>a</sup>	8.731 <sup>a</sup>	8.509 <sup>a</sup>	2.117 <sup>a</sup>	7/1/0
CPSO-H	2.728 <sup>a</sup>	4.929 <sup>a</sup>	8.458 <sup>a</sup>	7.962 <sup>a</sup>	3.510 <sup>a</sup>	N/A <sup>b</sup>	2.672 <sup>a</sup>	6.613 <sup>a</sup>	7/1/0
CLPSO	1.306 <sup>b</sup>	7.925 <sup>a</sup>	8.417 <sup>a</sup>	5.193 <sup>a</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>	-0.001 <sup>b</sup>	3/5/0
ABC	9.449 <sup>a</sup>	4.669 <sup>a</sup>	-15.481 <sup>c</sup>	9.491 <sup>a</sup>	1.071 <sup>b</sup>	1.696 <sup>a</sup>	9.184 <sup>a</sup>	0.393 <sup>b</sup>	5/2/1
Modified ABC	8.475 <sup>a</sup>	2.791 <sup>a</sup>	-16.022 <sup>c</sup>	6.504 <sup>a</sup>	N/A <sup>b</sup>	1.014 <sup>b</sup>	1.136 <sup>b</sup>	0.200 <sup>b</sup>	3/4/1
TLBO	N/A <sup>b</sup>	14.230 <sup>a</sup>	-1.870 <sup>c</sup>	N/A <sup>b</sup>	9.61E+015 <sup>a</sup>	1.008 <sup>b</sup>	1.180 <sup>b</sup>	5.587 <sup>a</sup>	3/4/1
I-TLBO	N/A <sup>b</sup>	7.708 <sup>a</sup>	-6.334 <sup>c</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>	4.019 <sup>a</sup>	2/5/1

<sup>a</sup> FATLBO is significantly better than the compared algorithm  
<sup>b</sup> FATLBO has no significant performance difference with the compared algorithm  
<sup>c</sup> FATLBO is significantly worse than the compared algorithm

**Table 5** Details of benchmark functions in Experiment 2 (*D*: dimensions)

No.	Function	Formulation	<i>D</i>	Search range
F1	De Jong	$F_{\min} = 3905.93 - 100(x_1^2 - x_2)^2 - (1 - x_1)^2$	2	[-2.048, 2.048]
F2	Goldstein-Price	$F_{\min} = \left[ 1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \left[ 30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \right]$	2	[-2, 2]
F3	Branin	$F_{\min} = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	2	[-5, 10]
F4	Martin and Gaddy	$F_{\min} = (x_1 - x_2)^2 + [(x_1 + x_2 - 10)/3]^2$	2	[0, 10]
F5	Rosenbrock1	$F_{\min} = 100(x_2 - x_1 + 1)^2 + (1 - x_1)^2$	2	[-1.2, 1.2]
F6	Rosenbrock2	$F_{\min} = 100(x_2 - x_1)^2 + (1 - x_1)^2$	2	[-10, 10]
F7	Rosenbrock3	$F_{\min} = \sum_{i=1}^D (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2)$	3	[-10, 10]
F8	Hyper Sphere	$F_{\min} = \sum_{i=1}^D x_i^2$	6	[-5.12, 5.12]

between the fitness value and the global solution is lower than 0.1 % or less than 0.001, whichever is smaller. FATLBO uses a population size of 40. Each benchmark function is tested 100 times, and Table 6 shows the comparative results in the form of mean of function evaluations and success percentages. The results of the other competing algorithms are obtained from previous works.

Table 5 demonstrates that the FATLBO algorithm is able to produce the fewest number of function evaluations in 7 out of 8 problems. Moreover, FATLBO requires approximately 29 and 52 % fewer function evaluations in

the De Jong function (F1) than I-TLBO and TLBO, respectively; 17 and 32 % in the Branin function (F3); 30 and 61 % in the Martin and Gaddy function (F4); 13 and 32 % in the Rosenbrock1 function (F5); 8 and 32 % in the Rosenbrock2 function (F6); 61 and 68 % in the Rosenbrock3 function (F7); and 14 and 23 % in the Hyper Sphere function (F8). The only function where FATLBO failed to achieve the fewest number of function evaluations is the Goldstein and Price function (F2). However, FATLBO still produces the second fewest number of function evaluations after I-TLBO.

**Table 6** Comparative results of TLBO and other algorithms in Experiment 2

	De Jong		Goldstein and price		Branin		Martin and Gaddy	
	MNFE	Success %	MNFE	Success %	MNFE	Success %	MNFE	Success %
SIMPSA	–	–	–	–	–	–	–	–
NE-SIMPSA	–	–	–	–	–	–	–	–
GA	10,160	100	5662	100	7325	100	2488	100
ANTS	6000	100	5330	100	1936	100	1688	100
BA	868	100	999	100	1657	100	526	100
GEM	746	100	701	100	689	100	258	100
TLBO	1070	100	452	100	443	100	422	100
I-TLBO	722	100	288	100	367	100	233	100
FATLBO	<i>514</i>	100	443	100	302	100	<i>163</i>	100
	Rosenbrock ( $D = 2$ )		Rosenbrock ( $D = 2$ )		Rosenbrock ( $D = 4$ )		Hyper Sphere ( $D = 6$ )	
SIMPSA	10,780	100	12,500	100	21,177	99	–	–
NE-SIMPSA	4508	100	5007	100	3053	94	–	–
GA	10,212	100	–	–	–	–	15,468	100
ANTS	6842	100	7505	100	8471	100	22,050	100
BA	631	100	2306	100	28,529	100	7113	100
GEM	572	100	2289	100	82,188	100	423	100
TLBO	669	100	1986	100	21,426	100	417	100
I-TLBO	522	100	964	100	17,696	100	376	100
FATLBO	<i>454</i>	100	887	100	<i>6841</i>	100	<i>321</i>	100

The number in italics indicates the best value

### 5.3 Experiment 3

In this experiment, the FATLBO algorithm is compared with TLBO and ABC variants in a small scale to a large scale by considering the dimensions 20, 30, and 50 for all the benchmark functions. In this part of the work, TLBO and I-TLBO are tested on 13 unconstrained benchmark functions, an approach that was first attempted by Li et al. [40] and later by Rao and Patel [27].

The complete formulation of these benchmark functions is given in Table 7. All functions have the global minimum 0 on every dimension except the Schwefel function (F8). The first seven functions (F1–F7) are categorized as unimodal, and the remainder (F8–F13) is multimodal.

Each benchmark function is tested 30 times with the FATLBO algorithm, and the results are obtained in the form of mean solution and standard deviation. The maximum number of function evaluations is 40,000. The population size of FATLBO is 10 for all functions except F5, F12, and F13, which use a population of 40.

Table 8 shows the comparative results of FATLBO, I-TLBO, TLBO, GABC, ABC, and I-ABC algorithms for 13 functions with the maximum of 40,000 function evaluations. Except for FATLBO, the rest of the results are obtained from the previous works. The experiment

revealed that the FATLBO algorithm outperformed the rest of the considered algorithms. FATLBO achieved the highest score in finding solutions more effectively than others (26) and surpassed the newly developed I-TLBO (19), I-ABC (17), and the original TLBO (9). It can be noted that FATLBO produces a better solution precision, especially in solving the Step function as well as the Quartic function. It is also possible to see that the success of FATLBO in solving the benchmark functions is not over-sensitive to the function dimension or the function characteristic. FATLBO performs well in various dimensions and unimodal or multimodal functions.

Table 9 shows the statistical results of FATLBO compared with those of other algorithms using an unpaired  $t$  test with a statistical significance value  $\alpha = 0.05$ . The last row of Table 9 presents the total count of the three statistically significant conclusions of performance comparison between FATLBO and the other algorithms from the  $t$  test ( $a/b/c$ ). According to the table, FATLBO outperforms the other algorithms, including the ABC and variants, the original TLBO, and the improved version of TLBO (I-TLBO). In comparison with FATLBO, the second best algorithm is I-TLBO where FATLBO has significantly better performance in 14 functions, statistically similar performance in 18 functions, and significantly worse performance in 7 functions.

**Table 7** Details of benchmark functions in Experiments 3 and 4 ( $D$ : dimensions)

No.	Function	Formulation	Search range
F1	Sphere	$F_{\min} = \sum_{i=1}^D x_i^2$	$[-100, 100]$
F2	Schwefel 2.22	$F_{\min} = \sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $	$[-10, 10]$
F3	Schwefel 1.2	$F_{\min} = \sum_{i=1}^D \left( \sum_{j=1}^i x_j^2 \right)^2$	$[-100, 100]$
F4	Schwefel 2.21	$F_{\min} = \max( x_i )$	$[-100, 100]$
F5	Rosenbrock	$F_{\min} = \sum_{i=1}^D [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$	$[-30, 30]$
F6	Step	$F_{\min} = \sum_{i=1}^D [ x_i + 0.5 ]^2$	$[-100, 100]$
F7	Quartic	$F_{\min} = \sum_{i=1}^D ix_i^4 + \text{rand}(0, 1)$	$[-1.28, 1.28]$
F8	Schwefel	$F_{\min} = -\sum_{i=1}^D (x_i \sin(\sqrt{ x_i }))$	$[-500, 500]$
F9	Rastrigin	$F_{\min} = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$
F10	Ackley	$F_{\min} = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i\right) + 20 + e$	$[-32, 32]$
F11	Griewank	$F_{\min} = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{ i }}\right) + 1$	$[-600, 600]$
F12	Penalized	$F_{\min} = \frac{\pi}{D} \left[ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 (1 + 10 \sin^2(\pi y_{i+1})) + (y_D - 1)^2 \right]$ $+ \sum_{i=1}^D u(x_i, 10, 100, 4), u(x_i, a, k, m)$ $= \begin{cases} k(x_i - a)^m & x_i > a, \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]$
F13	Penalized 2	$F_{\min} = 0.1 \left[ \sin^2(\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) + (x_D - 1)^2 + (1 + \sin^2(2\pi x_D)) \right]$ $+ \sum_{i=1}^D u(x_i, 5, 100, 4), u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a, \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]$

**Table 8** Comparative results of TLBO and others algorithm in Experiment 3

No.	Function	D	ABC		I-ABC		GABC		TLBO		I-TLBO		FATLBO		
			Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	
F1	Sphere	20	6.18E-16	2.11E-16	0	0	3.19E-16	7.39E-17	0	0	0	0	0	0	0
		30	3.62E-09	5.85E-09	0	0	6.26E-16	1.08E-16	0	0	0	0	0	0	0
		50	1.11E-05	1.25E-05	0	0	1.25E-05	6.05E-09	0	0	0	0	0	0	0
F2	Schwefel 2.22	20	1.35E-10	7.15E-11	0	0	9.36E-16	1.33E-16	0	0	0	0	0	0	0
		30	5.11E-06	2.23E-06	0	0	1.31E-10	4.69E-11	0	0	0	0	0	0	0
		50	2.92E-03	9.05E-04	0	0	2.37E-05	6.19E-06	0	0	0	0	0	0	0
F3	Schwefel 1.2	20	3.13E+03	1.19E+03	4.54E+03	2.69E+03	2.69E+03	1.46E+03	3.29E-38	1.20E-37	0	0	0	0	0
		30	1.24E+04	3.01E+03	1.43E+04	2.73E+03	1.09E+04	2.57E+03	3.25E-27	8.21E-27	0	0	0	0	0
		50	4.57E+04	6.46E+03	4.69E+04	7.36E+03	4.12E+04	5.83E+03	1.38E-21	4.00E-21	0	0	0	0	0
F4	Schwefel 2.21	20	3.9602	1.37E+00	0	0	0.3325	1.08E+00	7.19E-278	6.90E-278	0	0	0	0	0
		30	24.5694	5.66E+00	1.21E-197	0	12.6211	2.66E+00	3.96E-253	4.24E-253	4.7E-324	0	0	0	0
		50	56.338	4.84E+00	25.5055	5.67E+00	45.3075	4.32E+00	4.77E-234	5.11E-234	4.9E-324	0	0	0	0
F5	Rosenbrock	20	1.1114	1.80E+00	15.7165	1.40E+00	1.6769	2.90E+00	16.0706	3.68E-01	11.0955	8.71E-01	8.0270	1.23E+00	
		30	4.5509	4.88E+00	26.4282	1.40E+00	7.4796	1.91E+01	26.6567	2.94E-01	22.7934	5.82E-01	21.1703	1.06E+00	
		50	48.03	4.67E+01	47.028	8.60E-01	25.7164	3.18E+01	47.0162	3.56E-01	43.9786	4.55E-01	43.6530	1.02E+00	
F6	Step	20	5.55E-16	1.69E-16	6.31E-16	2.13E-16	3.34E-16	1.02E-16	1.99E-20	5.03E-20	6.16E-33	4.11E-33	0	0	
		30	2.49E-09	3.68E-09	3.84E-10	2.32E-10	6.45E-16	1.11E-16	2.74E-09	5.36E-09	1.17E-26	3.55E-26	0	0	
		50	1.36E-05	1.75E-05	1.84E-05	1.74E-05	5.65E-09	3.69E-09	6.26E-04	6.33E-04	1.39E-11	1.61E-11	0	0	
F7	Quartic	20	6.51E-02	2.03E-02	8.71E-03	3.24E-03	3.31E-02	7.93E-03	1.71E-02	1.01E-02	6.31E-03	6.45E-03	2.22E-04	1.38E-04	
		30	1.56E-01	4.65E-02	1.96E-02	9.34E-02	8.48E-02	2.79E-02	1.71E-02	8.95E-03	8.29E-03	3.30E-03	2.33E-04	1.64E-04	
		50	4.88E-01	1.07E-01	8.83E-02	2.55E-02	2.46E-01	4.72E-02	1.59E-02	8.11E-03	9.68E-03	3.88E-03	2.41E-04	1.41E-04	
F8	Schwefel	20	-8327.49	6.63E+01	-8323.77	7.40E+01	-8355.92	7.23E+01	-8105.47	1.74E+02	-8202.98	1.27E+02	-8379.43	3.17E-01	
		30	-12.130.31	1.59E+02	-12.251.03	1.67E+02	-12.407.29	1.06E+02	-12.311.72	2.21E+02	-12.351.40	1.35E+02	-12.566.66	2.50E+00	
		50	-19.326.5	2.66E+02	-19.313.49	2.77E+02	-19.975.29	2.31E+02	-20.437.84	1.48E+02	-20.533.71	2.46E+02	-20.927.70	2.53E+01	
F9	Rastrigin	20	1.41E-11	4.05E-11	0	0	0	0	1.95E-13	2.32E-13	0	0	0	0	
		30	0.4531	5.15E-01	0	0	0.0331	1.81E-01	1.87E-12	6.66E-12	0	0	0	0	
		50	8.4433	2.70E+00	0	0	2.1733	1.07E+00	2.03E-12	5.46E-12	0	0	0	0	
F10	Ackley	20	2.83E-09	2.58E-09	8.88E-16	0	2.75E-14	3.58E-15	3.55E-15	8.32E-31	7.11E-16	1.50E-15	4.32E-15	6.49E-16	
		30	2.75E-05	2.13E-05	8.88E-16	0	7.78E-10	2.98E-10	3.55E-15	8.32E-31	1.42E-15	1.83E-15	4.44E-15	9.33E-16	
		50	4.71E-02	3.40E-02	8.88E-16	0	1.11E-04	3.88E-05	3.55E-15	8.32E-31	1.42E-15	1.83E-15	4.20E-15	9.01E-16	
F11	Griewank	20	3.71E-03	6.61E-03	0	0	6.02E-04	2.23E-03	0	0	0	0	0	0	
		30	3.81E-03	8.45E-03	0	0	6.96E-04	2.26E-03	0	0	0	0	0	0	
		50	1.19E-02	1.97E-02	0	0	1.04E-03	2.74E-03	0	0	0	0	0	0	
F12	Penalized	20	4.06E-16	9.42E-17	4.17E-16	1.09E-16	3.26E-16	6.67E-17	1.13E-06	1.15E-06	4.00E-08	9.72E-15	1.26E-18	3.26E-18	
		30	1.18E-10	2.56E-10	7.10E-12	5.25E-12	5.86E-16	1.13E-16	6.16E-03	2.34E-02	2.67E-08	1.15E-13	3.74E-11	6.14E-11	
		50	8.95E-06	3.21E-05	5.42E-07	2.98E-07	9.30E-11	7.96E-11	6.01E-02	6.71E-02	5.72E-08	2.81E-08	3.83E-06	4.86E-06	
F13	Penalized 2	20	6.93E-08	2.92E-07	1.75E-16	4.54E-16	6.55E-08	2.44E-07	1.13E-06	1.15E-06	2.54E-08	3.77E-11	6.9E-09	2.61E-08	
		30	2.27E-07	4.12E-07	4.78E-08	2.04E-07	2.17E-07	5.66E-07	6.16E-03	2.34E-02	2.55E-08	4.89E-11	9.94E-04	2.12E-03	
		50	1.35E-05	2.78E-05	2.41E-05	4.35E-05	8.87E-07	1.53E-06	6.01E-02	6.71E-02	1.82E-06	1.08E-06	1.49E-03	3.81E-03	
		Count of solution better than others	3	17	5	9	19	26							

The number in italics indicates the best value

**Table 9** Statistical comparison of FATLBO with other optimization algorithms using an unpaired *t* test in Experiment 3

No.	Function	<i>D</i>	<i>t</i> value with $\alpha = 0.05$ and degree of freedom = 58				
			ABC	I-ABC	GABC	TLBO	I-TLBO
F1	Sphere	20	16.042 <sup>a</sup>	N/A <sup>b</sup>	23.643 <sup>a</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>
		30	3.389 <sup>a</sup>	N/A <sup>b</sup>	31.748 <sup>a</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>
		50	4.864 <sup>a</sup>	N/A <sup>b</sup>	11,316.582 <sup>a</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>
F2	Schwefel 2.22	20	10.342 <sup>a</sup>	N/A <sup>b</sup>	38.546 <sup>a</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>
		30	12.551 <sup>a</sup>	N/A <sup>b</sup>	15.299 <sup>a</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>
		50	17.672 <sup>a</sup>	N/A <sup>b</sup>	20.971 <sup>a</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>
F3	Schwefel 1.2	20	14.406 <sup>a</sup>	9.244 <sup>a</sup>	10.092 <sup>a</sup>	1.502 <sup>b</sup>	N/A <sup>b</sup>
		30	22.564 <sup>a</sup>	28.690 <sup>a</sup>	23.230 <sup>a</sup>	2.168 <sup>a</sup>	N/A <sup>b</sup>
		50	38.748 <sup>a</sup>	34.902 <sup>a</sup>	38.707 <sup>a</sup>	1.890 <sup>a</sup>	N/A <sup>b</sup>
F4	Schwefel 2.21	20	15.833 <sup>a</sup>	–Inf <sup>c</sup>	1.686 <sup>a</sup>	Inf <sup>a</sup>	–Inf <sup>c</sup>
		30	23.776 <sup>a</sup>	Inf <sup>a</sup>	25.988 <sup>a</sup>	Inf <sup>a</sup>	N/A <sup>b</sup>
		50	63.755 <sup>a</sup>	24.638 <sup>a</sup>	57.444 <sup>a</sup>	Inf <sup>a</sup>	N/A <sup>b</sup>
F5	Rosenbrock	20	–17.374 <sup>c</sup>	22.600 <sup>a</sup>	–11.041 <sup>c</sup>	34.315 <sup>a</sup>	11.151 <sup>a</sup>
		30	–18.228 <sup>c</sup>	16.400 <sup>a</sup>	–3.920 <sup>c</sup>	27.318 <sup>a</sup>	7.352 <sup>a</sup>
		50	0.513 <sup>b</sup>	13.856 <sup>a</sup>	–3.088 <sup>c</sup>	17.051 <sup>a</sup>	1.597 <sup>b</sup>
F6	Step	20	17.987 <sup>a</sup>	16.226 <sup>a</sup>	17.935 <sup>a</sup>	2.167 <sup>a</sup>	8.209 <sup>a</sup>
		30	3.706 <sup>a</sup>	9.066 <sup>a</sup>	31.827 <sup>a</sup>	2.800 <sup>a</sup>	1.805 <sup>a</sup>
		50	4.257 <sup>a</sup>	5.792 <sup>a</sup>	8.387 <sup>a</sup>	5.417 <sup>a</sup>	4.729 <sup>a</sup>
F7	Quartic	20	17.505 <sup>a</sup>	14.336 <sup>a</sup>	22.705 <sup>a</sup>	9.152 <sup>a</sup>	5.169 <sup>a</sup>
		30	18.348 <sup>a</sup>	11.356 <sup>a</sup>	16.602 <sup>a</sup>	10.321 <sup>a</sup>	10.255 <sup>a</sup>
		50	24.968 <sup>a</sup>	18.914 <sup>a</sup>	28.518 <sup>a</sup>	10.574 <sup>a</sup>	13.316 <sup>a</sup>
F8	Schwefel	20	4.291 <sup>a</sup>	4.120 <sup>a</sup>	1.781 <sup>a</sup>	8.624 <sup>a</sup>	7.610 <sup>a</sup>
		30	15.030 <sup>a</sup>	10.351 <sup>a</sup>	8.233 <sup>a</sup>	6.318 <sup>a</sup>	8.732 <sup>a</sup>
		50	32.822 <sup>a</sup>	31.786 <sup>a</sup>	22.448 <sup>a</sup>	17.870 <sup>a</sup>	8.726 <sup>a</sup>
F9	Rastrigin	20	1.907 <sup>a</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>	4.604 <sup>a</sup>	N/A <sup>b</sup>
		30	4.819 <sup>a</sup>	N/A <sup>b</sup>	1.002 <sup>b</sup>	1.538 <sup>b</sup>	N/A <sup>b</sup>
		50	17.128 <sup>a</sup>	N/A <sup>b</sup>	11.125 <sup>a</sup>	2.036 <sup>a</sup>	N/A <sup>b</sup>
F10	Ackley	20	6.008 <sup>a</sup>	–28.964 <sup>c</sup>	34.895 <sup>a</sup>	–6.498 <sup>c</sup>	–12.095 <sup>c</sup>
		30	7.072 <sup>a</sup>	–20.852 <sup>c</sup>	14.300 <sup>a</sup>	–5.225 <sup>c</sup>	–8.053 <sup>c</sup>
		50	7.588 <sup>a</sup>	–20.134 <sup>c</sup>	15.669 <sup>a</sup>	–3.951 <sup>c</sup>	–7.465 <sup>c</sup>
F11	Griewank	20	3.074 <sup>a</sup>	N/A <sup>b</sup>	1.479 <sup>b</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>
		30	2.470 <sup>a</sup>	N/A <sup>b</sup>	1.687 <sup>a</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>
		50	3.309 <sup>a</sup>	N/A <sup>b</sup>	2.079 <sup>a</sup>	N/A <sup>b</sup>	N/A <sup>b</sup>
F12	Penalized	20	23.519 <sup>a</sup>	20.882 <sup>a</sup>	26.635 <sup>a</sup>	5.382 <sup>a</sup>	2.26E+007 <sup>a</sup>
		30	1.677 <sup>a</sup>	–2.693 <sup>c</sup>	–3.336 <sup>c</sup>	1.442 <sup>b</sup>	2378.450 <sup>a</sup>
		50	0.864 <sup>b</sup>	–3.699 <sup>c</sup>	–4.316 <sup>c</sup>	4.906 <sup>a</sup>	–4.252 <sup>c</sup>
F13	Penalized 2	20	1.166 <sup>b</sup>	–1.448 <sup>b</sup>	1.308 <sup>b</sup>	5.348 <sup>a</sup>	3.882 <sup>a</sup>
		30	–2.568 <sup>c</sup>	–2.568 <sup>c</sup>	–2.568 <sup>c</sup>	1.204 <sup>b</sup>	–2.568 <sup>c</sup>
		50	–2.123 <sup>c</sup>	–2.107 <sup>c</sup>	–2.141 <sup>c</sup>	4.777 <sup>a</sup>	–2.139 <sup>c</sup>
Total (a/b/c)			32/3/4	18/13/8	28/4/7	23/13/3	14/18/7

<sup>a</sup> FATLBO is significantly better than the compared algorithm

<sup>b</sup> FATLBO has no significant performance difference with the compared algorithm

<sup>c</sup> FATLBO is significantly worse than the compared algorithm

### 5.4 Experiment 4

This experiment is intended to identify the performance of FATLBO in solving complex and high-dimension

problems. The same thirteen benchmark functions from Experiment 3 are tested using the FATLBO and TLBO algorithms, each with 500 and 1000 dimensions, which Shi et al. [41] originally solved. For 500-D functions, 2.5E6

**Table 10** Comparative results of FATLBO and other algorithms in Experiment 4

No	Function	No. of Dim's	SaNSDE mean	DECC-O mean	DECC-G mean	TLBO mean	FATLBO mean
F1	Sphere	500	2.41E−11	2.28E−21	6.33E−27	<i>0.00E+00</i>	<i>0.00E+00</i>
		1000	6.97E+00	1.77E−20	2.17E−25	<i>0.00E+00</i>	<i>0.00E+00</i>
F2	Schwefel 2.22	500	5.27E−02	3.77E−10	5.95E−15	<i>0.00E+00</i>	<i>0.00E+00</i>
		1000	1.24E+00	Inf	5.37E−14	<i>0.00E+00</i>	<i>0.00E+00</i>
F3	Schwefel 1.2	500	2.03E−08	2.93E−19	6.17E−25	<i>0.00E+00</i>	<i>0.00E+00</i>
		1000	6.43E+01	8.69E−18	3.71E−23	<i>0.00E+00</i>	<i>0.00E+00</i>
F4	Schwefel 2.21	500	4.07E+01	6.01E+01	4.58E−05	6.87E−196	<i>1.00E−323</i>
		1000	4.99E+01	7.92E+01	1.01E−01	7.23E−137	<i>1.00E−323</i>
F5	Rosenbrock	500	1.33E+03	6.64E+02	4.92E+02	5.27E+02	<i>4.73E+02</i>
		1000	3.31E+03	1.48E+03	9.87E+02	1.11E+03	<i>9.83E+02</i>
F6	Step	500	3.12E+02	<i>0.00E+00</i>	<i>0.00E+00</i>	<i>0.00E+00</i>	<i>0.00E+00</i>
		1000	3.93E+03	<i>0.00E+00</i>	<i>0.00E+00</i>	<i>0.00E+00</i>	<i>0.00E+00</i>
F7	Quartic	500	1.28E+00	1.04E+01	1.50E−03	9.28E−03	<i>3.32E−04</i>
		1000	1.18E+01	2.21E+01	8.40E−03	8.46E−03	<i>6.09E−04</i>
F8	Schwefel	500	−201,796.5	<i>−209,491</i>	<i>−209,491</i>	<i>−209,491</i>	<i>−209,491</i>
		1000	−372,991	<i>−418,983</i>	<i>−418,983</i>	<i>−418,983</i>	<i>−418,983</i>
F9	Rastrigin	500	2.84E+02	1.76E+01	<i>0.00E+00</i>	<i>0.00E+00</i>	<i>0.00E+00</i>
		1000	8.69E+02	3.12E+01	3.55E−16	<i>0.00E+00</i>	<i>0.00E+00</i>
F10	Ackley	500	7.88E+00	1.86E−11	9.13E−14	7.99E−15	<i>7.99E−15</i>
		1000	1.12E+01	4.39E−11	2.22E−13	7.99E−15	<i>7.99E−15</i>
F11	Griewank	500	1.82E−01	5.02E−16	4.40E−16	<i>0.00E+00</i>	<i>0.00E+00</i>
		1000	4.80E−01	2.04E−15	1.01E−15	<i>0.00E+00</i>	<i>0.00E+00</i>
F12	Penalized	500	2.96E+00	2.17E−25	4.29E−21	4.57E−01	9.53E−07
		1000	8.97E+00	1.08E−24	6.89E−25	6.21E−01	7.89E−09
F13	Penalized 2	500	1.89E+02	5.04E−23	5.34E−18	2.98E−01	9.36E−07
		1000	7.41E+02	4.82E−22	2.55E−21	4.47E−01	8.30E−08
Count of solution better than others			0	7	6	16	22

The number in italics indicates the best value

maximum function evaluations are used. For 1000-D functions, 5E6 maximum function evaluations are used. To ensure that the comparison is consistent, the same maximum number of function evaluations was used to test both TLBO and FATLBO algorithms.

Each benchmark function is tested 30 times with TLBO and FATLBO algorithms, and results are obtained in the form of mean value and standard deviation. The population of TLBO and FATLBO consists of 50 individuals. In this experiment, both algorithms are compared with DE variants. Except for the TLBO and FATLBO algorithms, the rest of the results are taken from those previous works. The comparative results of TLBO, FATLBO, and DE variants are shown in Table 10.

It can be seen from the results that the FATLBO algorithm has found 22 solutions that are better than others. Meanwhile, TLBO, DECC-G, DECC-C, and SaNSDE found only 16, 6, 7, and 0 solutions better than others, respectively. This shows that FATLBO is not only capable of handling low-dimension problems, but that it can also

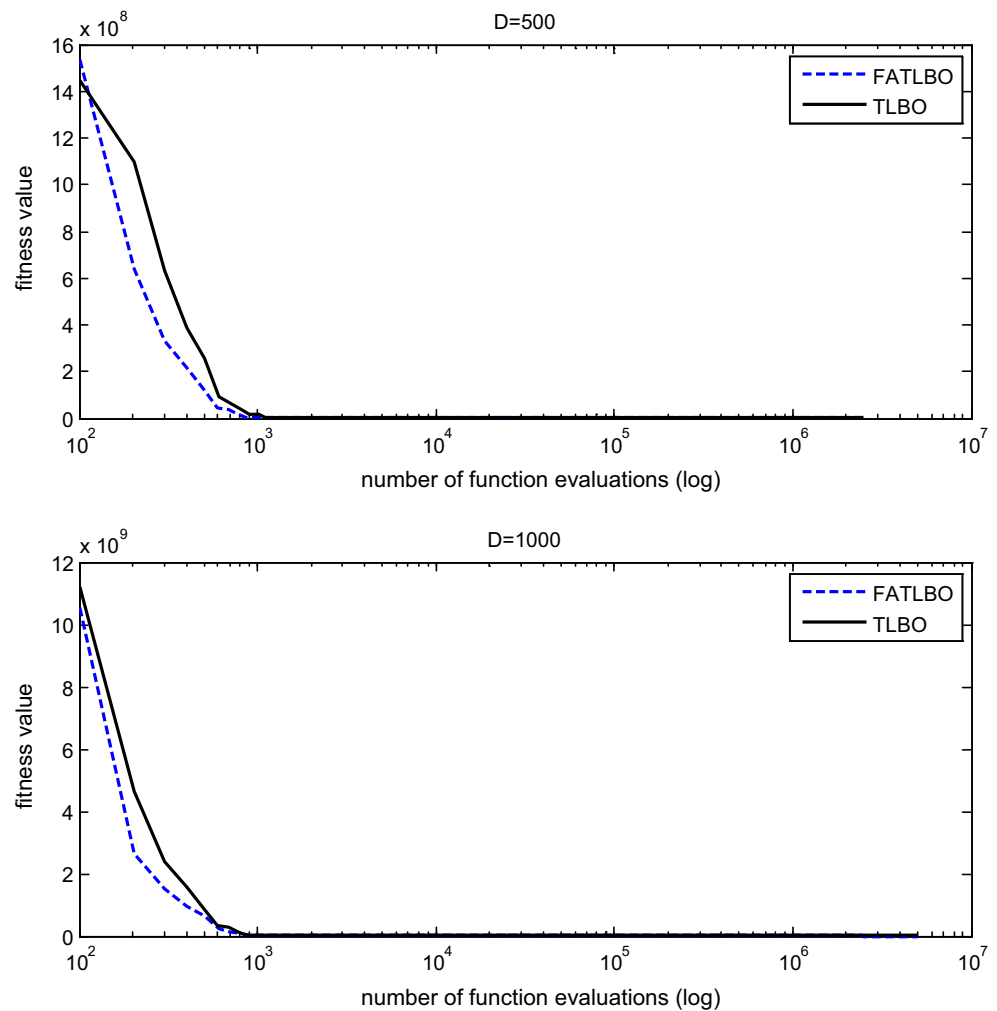
solve large-scale optimization problems. Meanwhile, FATLBO is able to outperform TLBO in F4, F5, F7, F12, and F13. For comparison purpose, the convergence graphics of TLBO and FATLBO algorithms for solving F13 are shown in Fig. 8. With the improved convergence characteristics provided by FATLS and a remedial operator, FATLBO is able to identify promising search regions faster than the regular TLBO.

## 6 Conclusion

This paper develops a new advanced metaheuristic algorithm so-called fuzzy adaptive teaching–learning-based optimization (FATLBO). FATLBO presents three new strategies to enhance the performance of TLBO, namely a status monitor, fuzzy adaptive teaching–learning strategies (FATLS), and a remedial operator. Two new parameters, teaching rate (TR) and learning rate (LR), are proposed to present the probability rate for a student to enter teacher



**Fig. 8** Convergence graphics of TLBO and FATLBO algorithms for F13 with  $D = 500$  and  $D = 1000$



and learner phases. According to the mathematical benchmark function results, it is proven that the FATLBO has a better performance than the TLBO, the ABC, the DE, and the PSO, as well as their variants. When it was applied to benchmark problems, the FATLBO had the capacity to generate much better solutions regarding the global optimization than other metaheuristic algorithms. Based on the obtained results, we conclude that the FATLBO has the capacity to achieve more optimal results with less function evaluations than previously tested algorithms.

The integration of three new modifications into an existing TLBO algorithm adds huge advantages to the searching capability of TLBO. The status monitor measures the productivity of students during the teacher and learner phases. FATLS adapts the current teaching–learning strategy into either a more teacher-centered orientation or more student-centered orientation based on the productivity information collected from the status monitor. The remedial operator applies random perturbation using Gaussian distribution principle when no fitness improvement occurs for a long time or the entire population

converges in one solution. The status monitor and FATLS increase the efficiency of knowledge transfer to each student and thus improve the optimization precision and convergence speed. Meanwhile, the remedial operator prevents the stagnation and avoids being trapped in local optima. We can conclude that the three new elements introduced here improve the quality of the results and increase the convergence speed of the basic TLBO algorithm. Thus, the novel FATLBO algorithm is proven to be a strong alternative for solving global optimization.

In addition, this paper defines a new understanding of parameter adaptation that may be described as one potential method for the parameter tuning process of FATLBO. More work is required to gather further evidence for possible shortcomings of the proposed algorithm. Meanwhile, optimizing the membership functions of each input variable can be a fruitful research agenda to further enhance the performance of the proposed algorithm. Finally, the application of the present fuzzy parameter adaptation to emerging metaheuristic algorithms like Jaya algorithm, SOS, and MVO could be a thought-provoking approach for future research.

## References

- Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Longman Publishing Co., Inc., New York
- Kennedy J, Eberhart R (1995) Particle swarm optimization. Paper presented at the proceedings of the IEEE international conference on neural networks, Perth, Australia
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359
- Mirjalili S, Mohd Hashim SZ, Moradian Sardroudi H (2012) Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Appl Math Comput* 218(22):11125–11137. doi:10.1016/j.amc.2012.04.069
- Cheng M-Y, Firdausi PM, Prayogo D (2014) High-performance concrete compressive strength prediction using Genetic Weighted Pyramid Operation Tree (GWPOT). *Eng Appl Artif Intell* 29:104–113. doi:10.1016/j.engappai.2013.11.014
- Cheng M-Y, Wibowo DK, Prayogo D, Roy AFV (2015) Predicting productivity loss caused by change orders using the evolutionary fuzzy support vector machine inference model. *J Civ Eng Manag* 21(7):881–892. doi:10.3846/13923730.2014.893922
- Sheikhan M, Ghoreishi SA (2012) Application of covariance matrix adaptation–evolution strategy to optimal control of hepatitis B infection. *Neural Comput Appl* 23(3):881–894. doi:10.1007/s00521-012-1013-3
- Arifovic J (1996) The behavior of the exchange rate in the genetic algorithm and experimental economies. *J Polit Econ* 104(3):510–541
- Cheng M-Y, Prayogo D, Wu Y-W (2014) Novel genetic algorithm-based evolutionary support vector machine for optimizing high-performance concrete mixture. *J Comput Civil Eng* 28(4):06014003. doi:10.1061/(asce)cp.1943-5487.0000347
- Talatahari S, Kheirollahi M, Farahmandpour C, Gandomi AH (2012) A multi-stage particle swarm for optimum design of truss structures. *Neural Comput Appl* 23(5):1297–1309. doi:10.1007/s00521-012-1072-5
- 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. doi:10.1007/s10898-007-9149-x
- Mirjalili S, Mirjalili SM, Hatamlou A (2015) Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl* 27(2):495–513. doi:10.1007/s00521-015-1870-7
- Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct* 139:98–112. doi:10.1016/j.compstruc.2014.03.007
- Rao RV (2016) Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int J Ind Eng Comput* 7(1):19–34. doi:10.5267/j.ijiec.2015.8.004
- Boussaïd I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. *Inform Sci* 237:82–117. doi:10.1016/j.ins.2013.02.041
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
- 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. doi:10.1016/j.cad.2010.12.015
- Rao RV, Savsani VJ, Vakharia DP (2012) Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Inform Sci* 183(1):1–15. doi:10.1016/j.ins.2011.08.006
- Satapathy S, Naik A (2011) Data clustering based on teaching–learning-based optimization. In: Panigrahi B, Suganthan P, Das S, Satapathy S (eds) *Swarm, Evolutionary, and Memetic Computing*, Lecture Notes in Computer Science, vol 7077. Springer, Berlin, pp 148–156. doi:10.1007/978-3-642-27242-4\_18
- Degertekin SO, Hayalioglu MS (2013) Sizing truss structures using teaching–learning-based optimization. *Comput Struct* 119:177–188. doi:10.1016/j.compstruc.2012.12.011
- 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. doi:10.1016/j.engappai.2012.06.007
- Niknam T, Azizpanah-Abarghoee R, Aghaei J (2013) A new modified teaching–learning algorithm for reserve constrained dynamic economic dispatch. *IEEE Trans Power Syst* 28(2):749–763
- Rao RV (2016) Teaching learning based optimization algorithm and its engineering applications. Springer International Publishing, Switzerland. doi:10.1007/978-3-319-22732-0
- Rao RV (2016) Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. *Decis Sci Lett* 5(1):1–30. doi:10.5267/j.dsl.2015.9.003
- Rao RV, Patel V (2012) An elitist teaching–learning-based optimization algorithm for solving complex constrained optimization problems. *Int J Ind Eng Comput* 3(4):535–560
- Satapathy SC, Naik A (2013) A modified teaching–learning-based optimization (mTLBO) for global search. *Recent Pat Comput Sci* 6(1):60–72
- Rao RV, Patel V (2012) An improved teaching–learning-based optimization algorithm for solving unconstrained optimization problems. *Sci Iran* 20(3):710–720. doi:10.1016/j.scient.2012.12.005
- Črepinšek M, Mernik M, Liu S-H (2011) Analysis of exploration and exploitation in evolutionary algorithms by ancestry trees. *Int J Innov Comput Appl* 3(1):11–19. doi:10.1504/IJICA.2011.037947
- Grefenstette JJ (1986) Optimization of control parameters for genetic algorithms. *IEEE Trans Syst Man Cybern* 16(1):122–128. doi:10.1109/TSMC.1986.289288
- Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput* 13(2):398–417. doi:10.1109/TEVC.2008.927706
- Yang I-T, Hsieh Y-H (2011) Reliability-based design optimization with discrete design variables and non-smooth performance functions: AB-PSO algorithm. *Autom Constr* 20(5):610–619. doi:10.1016/j.autcon.2010.12.003
- Shi Y, Eberhart RC. Fuzzy adaptive particle swarm optimization. In: *Proceedings of the 2001 congress on evolutionary computation, 2001*, vol 101, pp 101–106. doi:10.1109/CEC.2001.934377
- Melin P, Olivas F, Castillo O, Valdez F, Soria J, Valdez M (2013) Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. *Expert Syst Appl* 40(8):3196–3206. doi:10.1016/j.eswa.2012.12.033
- Liu J, Lampinen J (2004) A fuzzy adaptive differential evolution algorithm. *Soft Comput* 9(6):448–462. doi:10.1007/s00500-004-0363-x
- Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10(3):281–295. doi:10.1109/tevc.2005.857610

36. Akay B, Karaboga D (2012) A modified Artificial Bee Colony algorithm for real-parameter optimization. *Inform Sci* 192:120–142. doi:[10.1016/j.ins.2010.07.015](https://doi.org/10.1016/j.ins.2010.07.015)
37. Mathur M, Karale SB, Priye S, Jayaraman VK, Kulkarni BD (2000) Ant colony approach to continuous function optimization. *Ind Eng Chem Res* 39(10):3814–3822. doi:[10.1021/ie990700g](https://doi.org/10.1021/ie990700g)
38. Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M (2006) The Bees Algorithm—a novel tool for complex optimisation problems. Paper presented at the proceedings of the 2nd international virtual conference on intelligent production machines and systems (IPROMS 2006), Oxford
39. Ahrari A, Atai AA (2010) Grenade explosion method—a novel tool for optimization of multimodal functions. *Appl Soft Comput* 10(4):1132–1140. doi:[10.1016/j.asoc.2009.11.032](https://doi.org/10.1016/j.asoc.2009.11.032)
40. 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. doi:[10.1016/j.asoc.2011.08.040](https://doi.org/10.1016/j.asoc.2011.08.040)
41. Shi YJ, Teng HF, Li ZQ (2005) Cooperative co-evolutionary differential evolution for function optimization. In: Wang L, Chen K, Ong YS (eds) *Advances in natural computation. Lecture notes in computer science*, vol 3611. Springer, Berlin, pp 1080–1088. doi:[10.1007/11539117\\_147](https://doi.org/10.1007/11539117_147)