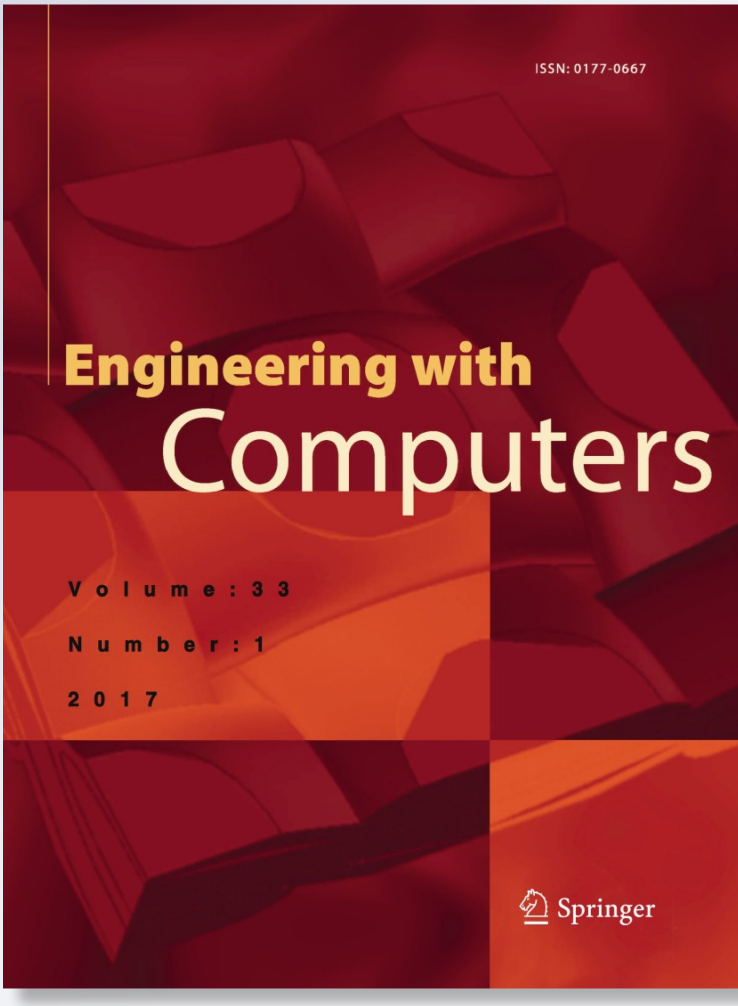
A novel fuzzy adaptive teaching–learning-based optimization (FATLBO) for solving structural optimization problems

Min-Yuan Cheng & Doddy Prayogo

Engineering with Computers
An International Journal for Simulation-Based Engineering

ISSN 0177-0667
Volume 33
Number 1

Engineering with Computers (2017)
33:55-69
DOI 10.1007/s00366-016-0456-z



Springer

Engineering with Computers (2017) 33:55–69
DOI 10.1007/s00366-016-0456-z

ORIGINAL ARTICLE

A novel fuzzy adaptive teaching–learning-based optimization (FATLBO) for solving structural optimization problems

Min-Yuan Cheng[1] **· Doddy Prayogo**[1,2]

Received: 1 March 2016 / Accepted: 19 May 2016 / Published online: 6 June 2016

Abstract This paper presents a new optimization algorithm called fuzzy adaptive teaching–learning-based optimization (FATLBO) for solving numerical structural problems. This new algorithm introduces three new mechanisms for increasing the searching capability of teaching–learning-based optimization namely status monitor, fuzzy adaptive teaching–learning strategies, and remedial operator. The performance of FATLBO is compared with well-known optimization methods on 26 unconstrained mathematical problems and five structural engineering design problems. Based on the obtained results, it can be concluded that FATLBO is able to deliver excellence and competitive performance in solving various structural optimization problems.

Keywords Optimization · Fuzzy logic · Teaching–learning-based optimization · Structural design problems

1 Introduction

Structural optimization is a challenging area of study that has attracted increasing attention in recent decades. Various gradient-based optimization methods have been developed to solve various engineering optimization problems. Most use analytical or numerical methods that require gradient information to improve initial solutions. However, gradient-based optimization methods are inadequate to resolve the complexities inherent in many of today's real-world structural design problems. Moreover, gradient search in problems with greater than one local optimum is difficult and unstable [1]. Shortcomings in current gradient-based approaches to engineering optimization have thus encouraged researchers to develop better optimization methods.

Metaheuristic is a research field which simulates different natural phenomena to solve a wide range of optimization problems. Researchers had proposed a number of algorithms in the past considering different natural phenomena including: genetic algorithm (GA) [2], particle swarm optimization (PSO) [3], differential evolution (DE) [4], symbiotic organisms search (SOS) [5, 6], multi-verse optimizer (MVO) [7], and jaya algorithm [8]. The teaching–learning-based optimization (TLBO) algorithm was first introduced by Rao et al. to solve various unconstrained [9] and constrained problems [10]. It is one of the newest metaheuristic techniques motivated by the school teaching–learning process in a classroom. In the TLBO algorithm, a student is a member in the classroom, representing a potential solution to the optimization process and a teacher is defined as the best solution in the classroom representing the best learner in the society. Each learner of the classroom group adjusts search patterns according to the knowledge transfer from a teacher first (teacher phase) and then from interaction with other fellow students (learner phase). One of the advantages over most metaheuristic algorithms is that TLBO uses only common control parameters like population size and number of generations, while other algorithms require additional algorithm-specific control parameters [9, 10].

✉ Doddy Prayogo
doddyprayogo@ymail.com

Min-Yuan Cheng
myc@mail.ntust.edu.tw

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

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

Since its inception, TLBO has been used in various applications. Some are applied in data clustering [11, 12], mechanical engineering [13, 14], power and energy systems [15–19], and structural engineering [20, 21]. More complete review on applications related to the TLBO can be found in the literature [22, 23]. As a new research, many studies also have been conducted to improve the performance of TLBO [24, 25]. All of these studies have been proven to enhance the performance of the original TLBO and thus the efforts must be continued to develop a better optimization method as many real-world problems become increasingly complex.

This paper introduces a novel algorithm based on TLBO namely fuzzy adaptive teaching–learning-based optimization (FATLBO) which offers significant performance improvement over its predecessor. Three major modifications are introduced: status monitor, fuzzy adaptive teaching–learning strategies (FATLS) and remedial operator. Status monitor is used to monitor the performance of the students in both teacher phase and learner phase. In FATLS, a mechanism is deployed to monitor the performance of both teacher phase and learner phase then fuzzy set theory is employed to emphasize the most productive phase resulting in increasing algorithm's convergence speed. Remedial operator is introduced as a restart mechanism when all learners do not make significant progress for a period of iteration. FATLBO is compared with a number of algorithms including the original TLBO, and other well-known metaheuristic algorithms in several experiments of benchmark functions and real case studies in structural engineering. The results of the experiments show that the proposed algorithms outperform the competing algorithms in solving most of the function problems considered in this paper.

The rest of the paper is organized as follows: Sect. 2 describes the standard TLBO algorithm. Section 3 presents the proposed FATLBO algorithm. Section 4 presents the experimental results. Finally, conclusions are given in Sect. 5.

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

The main idea behind TLBO is the simulation of a classical school learning process that consists of two stages. During the first stage, called teacher phase, a teacher imparts knowledge directly to his/her students. The better the teacher, the more knowledge the students obtain. However, the possibility of a teacher's teaching being successful during the teacher phase, in practice, is distributed under Gaussian law. There are only very rare students who can understand all the materials presented by the teacher (i.e., the right end of the Gaussian distribution). Most students will partially accept new learning materials (i.e., the mid part of the Gaussian distribution) and, in some cases, the teacher will have almost no direct effect on students' knowledge (i.e., the left end of the Gaussian distribution). However, the possibility for most students to obtain new knowledge is not completely lost. During the second stage, called learner phase, a student may learn with the help of fellow students. Overall, how much knowledge is transferred to a student does not only depend on his/her teacher but also on interactions amongst students through peer learning.

Like many metaheuristic algorithms, TLBO algorithm uses a population of students to search the global solution. An initial population is randomly generated according to the population size and number of design variables. A student (X_i) within the population indicates one possible solution to a particular optimization problem. X_i is a vector of design variables that represents the number of subjects offered to the students. Meanwhile, the student's result is represented by the fitness value vector of the optimization problem.

During the teacher phase, the smartest student with minimum fitness value is assigned as the teacher (X_{teacher}) for that iteration. TLBO attempts to improve other students (X_i) by shifting students' mean value (X_{mean}) towards the X_{teacher} as shown from Eq. (1). It can be seen that the student improvement may be influenced by the difference between the teacher's knowledge and the average knowledge of all students.

$$X_{i\ \text{new}} = X_i + \text{rand} \times (X_{\text{teacher}} - T_{\text{F}} \times X_{\text{mean}}) \tag{1}$$

where rand ranges between 0 and 1 and T_{F} is a teaching factor, which can be either 1 or 2.

During the learner phase, a student (X_i) tries increase the knowledge by the interaction between himself and another student 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, X_i is shifted towards X_j as shown in Eq. (2). Otherwise, it is shifted away from X_j as shown in Eq. (3).

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

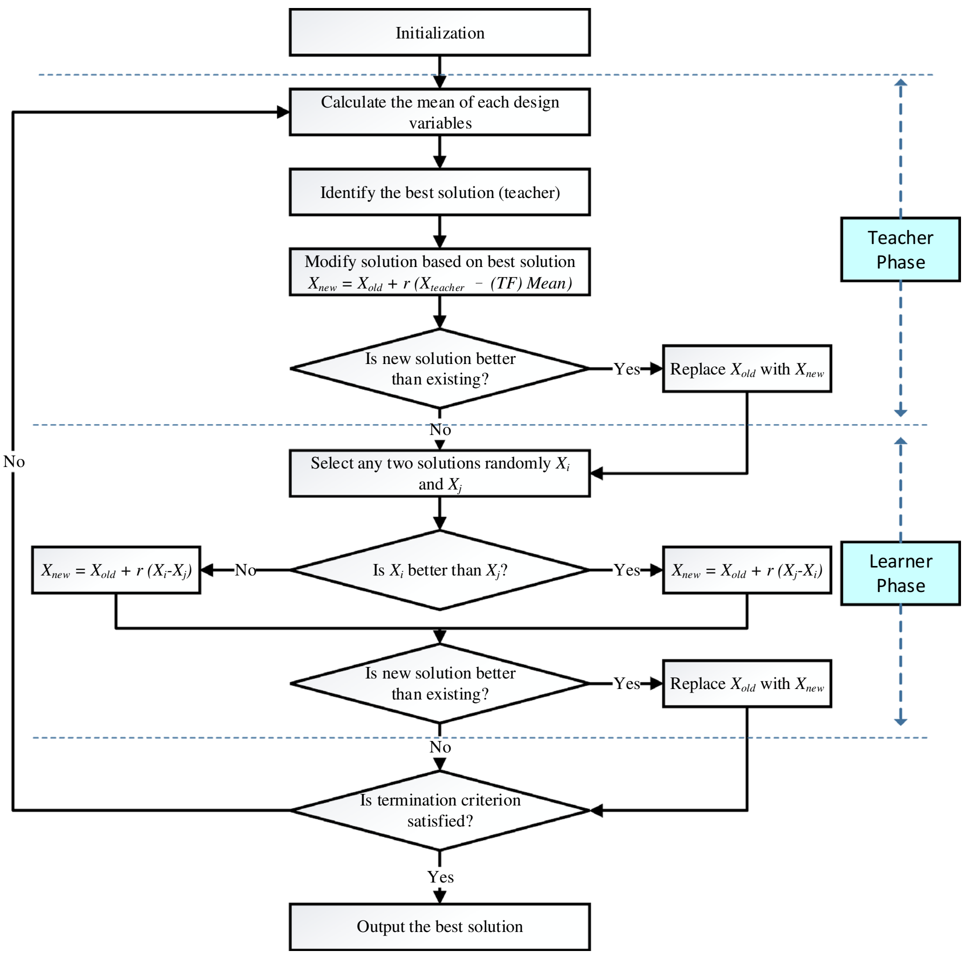
$$X_{i\ \text{new}} = X_i + \text{rand} \times (X_i - X_j) \tag{3}$$

For both teacher and learner phases, a student is only updated if the new solution ($X_{i\ \text{new}}$) is better than the previous one (X_i). TLBO then tries to improve certain students by changing these students during the teacher and learner phases, The algorithm will operate until reaching the maximum number of iterations. Figure 1 illustrates the flowchart of TLBO algorithm.

3 The fuzzy adaptive teaching–learning-based optimization (FATLBO)

In real-life, teaching method plays a critical role in the efficiency of transfer knowledge between teacher and students

Fig. 1 TLBO flowchart



in the classroom. For some materials, it will be more suitable to use the teacher-centered approach where the teacher dominantly gives a direct instruction to the students. However, for some topics, it might be more appropriate if the students are actively engaged in discussion and cooperative learning between themselves. Improper use of the teaching method will produce a negative impact on the learning process.

In the basic version, TLBO algorithm forces all students to enter both teacher and learner phases. As a result, TLBO requires total two function evaluations to calculate the fitness of every candidate solution created in one iteration. This paper implements the use of real-life teaching methods by introducing several modifications on the basic TLBO. In Sect. 3.1, a status monitor is developed to keep a track of the progress of students in each phase. Status monitor measures the productivity of each phase in improving the student's knowledge for a certain period of iteration. In Sect. 3.2, a new strategy is developed namely fuzzy adaptive teaching–learning system (FATLS) to decide the proper teaching methods for optimizing the searching speed and the efficiency of the TLBO algorithm. The students are allowed to skip teacher phase or learner phase by introducing two new parameters called teaching rate (TR) and learning rate (LR). TR and LR present the probability rate for a student to enter teacher and learner phase. If TR is set to 0.8, every student will have 80 % possibility to go to the teacher phase and 20 % possibility to skip the teacher phase. Fuzzy logic is further employed to self-adjust the TR and LR based on the information collected from the status monitor. Finally, remedial operator is introduced to ensure the students not trapped in local optima in Sect. 3.3.

3.1 Status monitor

Two new variables are introduced to measure the productivity of the students during teacher phase and learner phase namely the success rate of teacher phase (SRTP) and the success rate of learner phase (SRLP). The value of SRTP and SRLP are varied between 0 and 1 where the higher value means the more success of the certain phase to improve the students' knowledge.

Table 1 The 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	4.18	*3.28*	*Yes*	3.28	No
2	19.97	*10.1*	*Yes*	10.1	No
3	78.69	78.69	No	*66.79*	*Yes*
4	91.95	*35.18*	*Yes*	*28.79*	*Yes*
5	94.89	94.89	No	94.89	No
Success rate			SRTP = 0.6		SRLP = 0.4

The number in italics indicates the updated fitness value

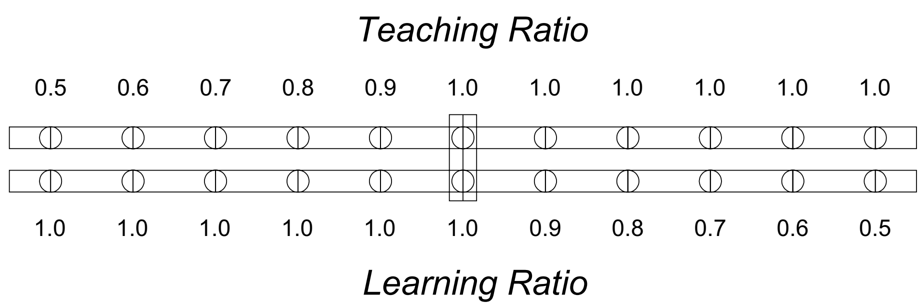


Fig. 2 Bar controller

Table 1 illustrates the example of SRTP and SRLP calculation in one iteration. According to the result, the SRTP is 0.6 while SRLP is 0.4. From the numbers, it can be interpreted that entering teacher phase will approximately improve the fitness of six-tenths of total students while only four-tenths of students success to improve the fitness during the learner phase. Status monitor will track the progress of the students in every iteration. The information of SRTP and SRLP produced by status monitor will be used by FATLS to adjust the fittest TR and LR for each student in the next step.

3.2 Fuzzy adaptive teaching–learning strategies (FATLS)

To boost the productivity of learning process, a fuzzy system is developed. The SRTP and SRLP obtained from the status monitor serve as the input variables. As for the output variables, bar controller is developed to adjust the TR and LR. The illustration of bar controller can be seen in Fig. 2. If the middle bar moves to the left, it will decrease the TR. In other words, the FATLS will emphasize the learner phase. If the bar moves to the right, it will decrease the LR and vice versa. The movement of bar controller will be used as the output variable. Then, the movement of bar controller is adjusted to tune the TR and LR.

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

3.2.1 Fuzzification

SRTP and SRLP were calculated previously in order to measure the productivity of the students during each phase. Hence, each SRTP and SRLP from the previous step are converted to the membership grade as shown in Fig. 3. The membership functions used for each input variable are left-triangle, triangle, and right-triangle membership functions. For output variable, five triangular membership functions are used including left-triangle, three triangles, and right-triangle as shown in Fig. 4.

3.2.2 Inferencing process

The Mamdani-type fuzzy rule is used to formulate the conditional statements that consist of fuzzy logic. Mamdani's fuzzy inference is used for mapping the given inputs to the output. There are nine fuzzy rules which are suggested as follows.

Rule 1 **IF** success rate of teacher phase is LOW **AND** success rate of learner phase is LOW; **THEN** bar movement is NEUTRAL.

Rule 2 **IF** success rate of teacher phase is LOW **AND** success rate of learner phase is MEDIUM; **THEN** bar movement is LEFT.

Rule 3 **IF** success rate of teacher phase is LOW **AND** success rate of learner phase is HIGH; **THEN** bar movement is FAR-LEFT.

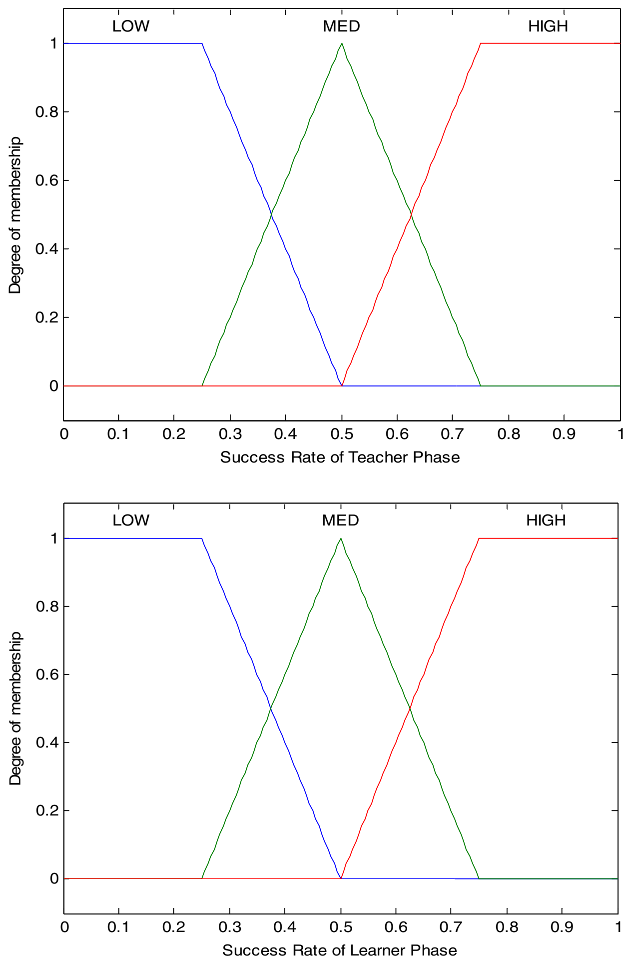


Fig. 3 Membership function of the SRTP and SRLP

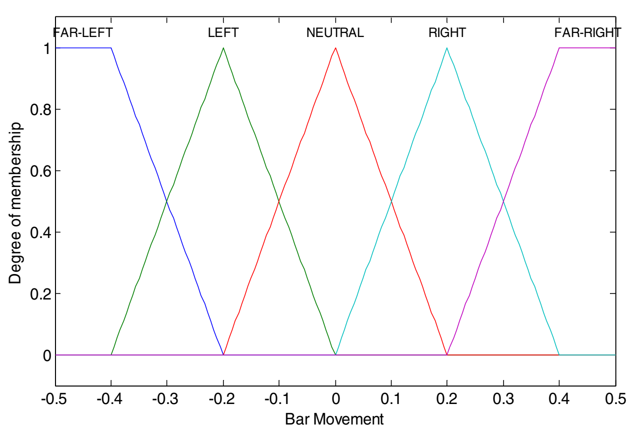


Fig. 4 Membership function of the bar movement

Rule 4 **IF** success rate of teacher phase is MEDIUM **AND** success rate of learner phase is LOW; **THEN** bar movement is RIGHT.

Rule 5 **IF** success rate of teacher phase is MEDIUM **AND** success rate of learner phase is MEDIUM; **THEN** bar movement is NEUTRAL.

Rule 6 **IF** success rate of teacher phase is MEDIUM **AND** success rate of learner phase is HIGH; **THEN** bar movement is LEFT.

Rule 7 **IF** success rate of teacher phase is HIGH **AND** success rate of learner phase is LOW; **THEN** bar movement is FAR-RIGHT.

Rule 8 **IF** success rate of teacher phase is HIGH **AND** success rate of learner phase is MEDIUM; **THEN** bar movement is RIGHT.

Rule 9 **IF** success rate of teacher phase is HIGH **AND** success rate of learner phase is HIGH; **THEN** bar movement is NEUTRAL.

3.2.3 Defuzzification

This step is a reverse step of fuzzification. Once the inferencing process finishes, the defuzzification begins. The center-of-sums method is selected as the defuzzification method. The output of the fuzzy strategies is the bar movement. Based on bar movement, the TR and LR can be adjusted for the next iterations. For example, if bar movement result is +0.26, the bar will move 2.6 step to the right. It means the TR and LR will be updated to 1.0 and 0.74, respectively (see Fig. 5).

3.3 Remedial operator

In the classroom, there is a situation when entire students have made a little improvement in the learning process. It might occur due to various things such as teaching style, heavy topic, or bad environment. Facing this problem, teacher might hold evaluation and add perturbation or remediation to the class based on the current situation. This remediation can be either changing the teaching style, applying personal tutoring, or downgrading the lessons loads.

As for FATLBO, when premature convergence occurred, remedial operator is changing the classroom environment by expanding the population using the random movement based on Gaussian distribution with the teacher as the center point. In this paper, the premature convergence is assumed when no fitness improvement for long times or entire population converges in one solution. The corresponding formula is shown below.

$$X_{i\ \mathrm{new}} = X_{\mathrm{teacher}} + N(0, \mathrm{range}/2),$$
where range is equal to upper bound and lower bound (4)

Finally, Fig. 6 summarizes the whole procedure of FATLS.

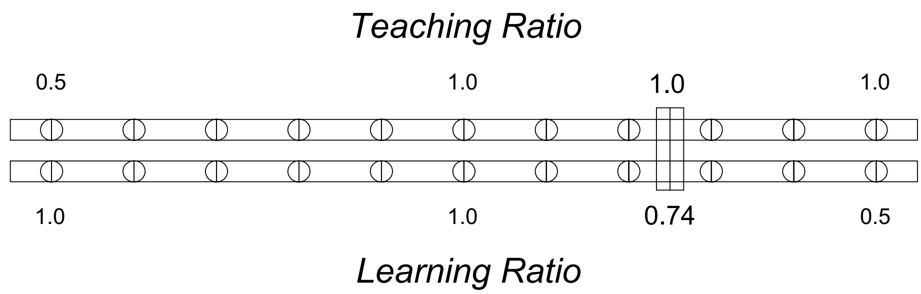


Fig. 5 Bar controller after applying FATLS

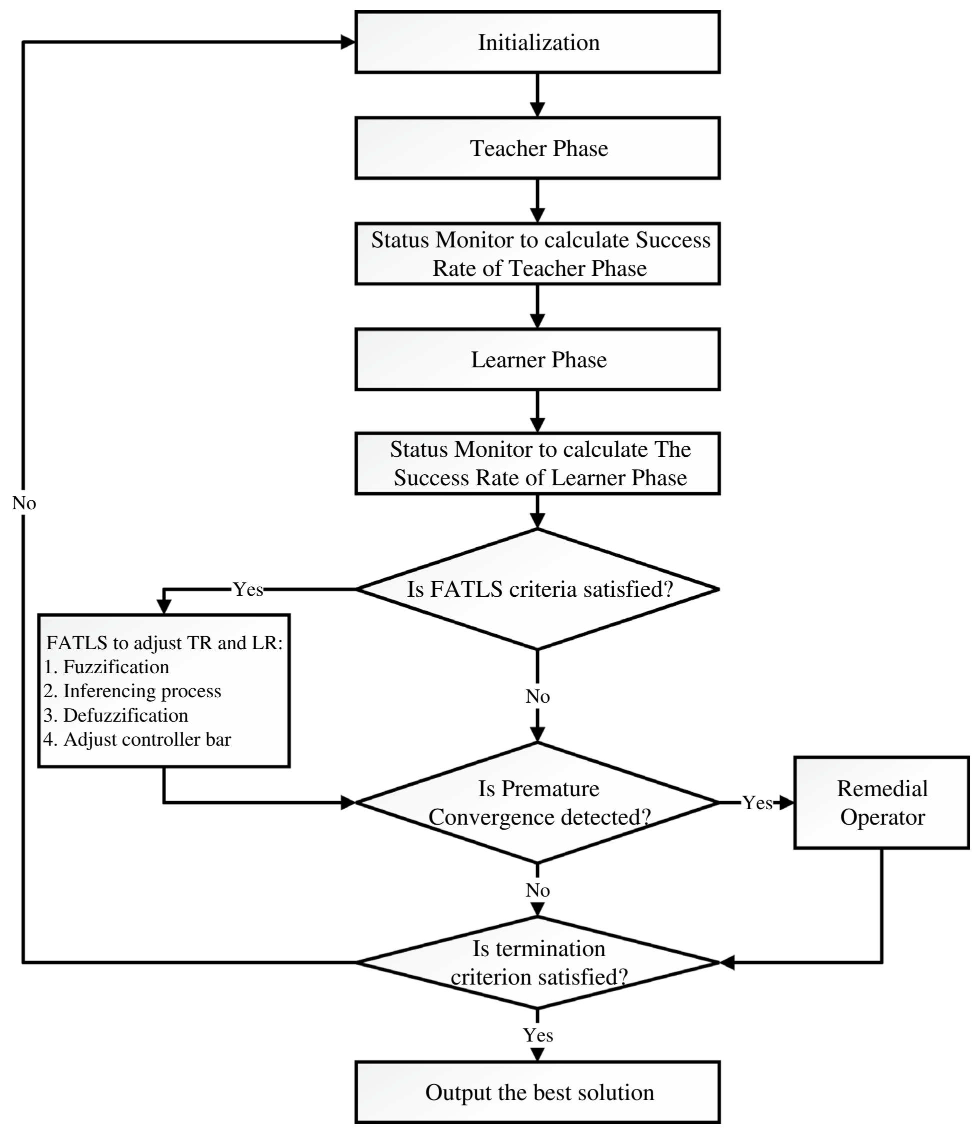


Fig. 6 FATLBO flowchart

4 FATLBO validation

This paper considered several numerical optimization problems from the literature to validate FATLBO performance. This section is divided into two sub-sections. Section 4.1 provides a large set of complex mathematical benchmark problems to be tested, with results compared against other metaheuristic algorithms. Section 4.2 examines five structural engineering design problems.

Table 2 The detailed of benchmark functions (D: dimensions, M: multimodal, N: non-Separable, U: unimodal, S: separable)

Number	Function	Range	D	Type	Formulation	Min
1	Beale	[−4.5, 4.5]	2	UN	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	0
2	Easom	[−100, 100]	2	UN	$f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2 - (x_2-\pi)^2)$	−1
3	Matyas	[−10, 10]	2	UN	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	0
4	Bohachevsky1	[−100, 100]	2	MS	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	0
5	Booth	[−10, 10]	2	MS	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	0
6	Michalewicz2	[0, π]	2	MS	$f(x) = -\sum_{i=1}^{D} \sin(x_i)(\sin(ix_i^2/\pi))^{20}$	−1.8013
7	Schaffer	[−100, 100]	2	MN	$f(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2+x_2^2}) - 0.5}{(1+0.001(x_1^2+x_2^2))^2}$	0
8	Six Hump Camel Back	[−5, 5]	2	MN	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	−1.03163
9	Boachevsky2	[−100, 100]	2	MN	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)(4\pi x_2) + 0.3$	0
10	Boachevsky3	[−100, 100]	2	MN	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	0
11	Shubert	[−10, 10]	2	MN	$f(x) = \left(\sum_{i=1}^{5} i\cos(i+1)x_1 + i\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$	−186.73
12	Colville	[−10, 10]	4	UN	$f(x) = 100(x_1^2 - x_2)^2 + (x_1-1)^2 + (x_3-1)^2 + 90(x_3^2 - x_4)^2 + 10.1(x_2-1)^2 + (x_4-1)^2 + 19.8(x_2-1)(x_4-1)$	0
13	Michalewicz5	[0, π]	5	MS	$f(x) = -\sum_{i=1}^{D} \sin(x_i)(\sin(ix_i^2/\pi))^{20}$	−4.6877
14	Zakharov	[−5, 10]	10	UN	$f(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^4$	0
15	Michalewicz10	[0, π]	10	MS	$f(x) = -\sum_{i=1}^{D} \sin(x_i)(\sin(ix_i^2/\pi))^{20}$	−9.6602
16	Step	[−5.12, 5.12]	30	US	$f(x) = \sum_{i=1}^{D} (x_i + 0.5)^2$	0
17	Sphere	[−100, 100]	30	US	$f(x) = \sum_{i=1}^{D} x_i^2$	0
18	SumSquares	[−10, 10]	30	US	$f(x) = \sum_{i=1}^{D} ix_i^2$	0
19	Quartic	[−1.28, 1.28]	30	US	$f(x) = \sum_{i=1}^{D} ix_i^4 + \text{Rand}$	0
20	Schwefel 2.22	[−10, 10]	30	UN	$f(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	0
21	Schwefel 1.2	[−100, 100]	30	UN	$f(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$	0
22	Rosenbrock	[−30, 30]	30	MN	$f_5(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	0
23	Dixon-Price	[−10, 10]	30	UN	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_i - 1)^2$	0
24	Rastrigin	[−5.12, 5.12]	30	MS	$f(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	0
25	Griewank	[−600, 600]	30	MN	$f(x) = \frac{1}{4000}\left(\sum_{i=1}^{D} (x_i - 100)^2\right) - \left(\prod_{i=1}^{D} \cos(\frac{x_i - 100}{\sqrt{i}})\right) + 1$	0
26	Ackley	[−32, 32]	30	MN	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	0

4.1 Mathematical benchmark problems

This section compares the performance of FATLBO to the performance of other metaheuristic algorithms including GA, DE, PSO, BA, and the hybrid PSO-BA [particle bee algorithm (PBA)] using 26 benchmark functions described by Cheng and Lien [26]. Functions 1–11 are two-dimensional; functions 12 and 13 are four- and

Table 3 Parameter settings of the algorithms

GA	DE	PSO	BA	PBA	TLBO	FATLBO
$n = 50$	$n = 50$	$n = 50$	$n = 50$	$n = 50$	$n = 50$	$n = 50$
$m = 0.01$	$c = 0.9$	$w = 0.9 \sim 0.7$	$e = \mathrm{NP}/2$	$e = \mathrm{NP}/2$		
$c = 0.8$	$F = 0.5$	$v = X_{\min}/10 \sim X_{\max}/10$	$b = \mathrm{NP}/4$	$b = \mathrm{NP}/4$		
$g = 0.9$			$r = \mathrm{NP}/4$	$r = \mathrm{NP}/4$		
			$n_1 = 2$	$w = 0.9 \sim 0.7$		
			$n_2 = 1$	$v = X_{\min}/10 \sim X_{\max}/10$		
				$Pelite = 15$		
				$Pbest = 9$		

n population size/colony size/ecosystem size, *m* mutation rate, *c* crossover rate, *g* generation gap, *f* scaling factor, *w* inertia weight, *v* limit of velocity, *e* elite bee number, *b* best bee number, *r* random bee number, n_1 elite bee neighborhood number, n_2 best bee neighborhood number, *Pelite* PSO iteration of elite bees, *Pbest* PSO iteration of best bees

five-dimensional; and functions 16–26 are 30-dimensional. All functions may be separated into the type categories of multimodal/unimodal and separable/non-separable. Table 2 provides benchmark function details.

Cheng and Lien [26] previously conducted experiments on all algorithms with a 500,000 maximum number of function evaluations for each benchmark functions. Any value less than 1E−12 was reported as 0. To maintain comparison consistency, FATLBO was also tested using these same conditions and number of function evaluations. Table 3 lists control and specific parameter settings for each algorithm.

Table 4 delineates the respective performance of FATLBO and other algorithms in solving benchmark functions. Performance values for all algorithms except for FATLBO reference Cheng and Lien [26]. The mean value and standard deviation for FATLBO were obtained after 30 independent runs, in line with standards followed in the previous work. In Table 4, bolded numbers represent the comparatively best values. FATLBO found the global optimum value for 23 of the 26 functions and outperformed all other algorithms tested. Further, FATLBO was the only algorithm able to solve Dixon-Price (function 23) and produced the best result of all on the exceptionally difficult Rosenbrock (function 22).

4.2 Structural design problems

This section examines FATLBO performance using five structural design optimization problems from the structural engineering field, with FATLBO optimization results compared to data published in the literature. FATLBO used 30 individuals and 30 independent runs for all cases. Different maximum numbers of function evaluations were used for each problem, with smaller function evaluation numbers used for smaller number of design variables and moderate functions and larger function evaluation numbers used for larger design variable numbers and complex problems. As for constraint handling method, Deb's feasibility rules is considered in this paper [27]. Because FATLBO is parameter-free, only common control parameters (population size and maximum number of function evaluations) were adjusted. Therefore, FATLBO performance was consistent across different problems.

4.2.1 Cantilever beam

The cantilever beam problem was adopted from Chickermane and Gea [28]. The cantilever beam shown in Fig. 7 comprises five elements. Each element has a hollow cross section of a fixed diameter. The beam is rigidly supported as shown, and a vertical force acts at the free end of the cantilever. The problem presented is to minimize beam weight. The design variable is the height (or width) x_i of each beam element. Bound constraints are set as $0.01 \le x_i \le 100$. The problem is formulated using classical beam theory as:

Minimize

$$f(X) = 0.0624\ (x_1 + x_2 + x_3 + x_4 + x_5) \quad (5)$$

Subject to:

$$g(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \le 1 \quad (6)$$

Table 5 lists the best solutions obtained by FATLBO and various methods [28, 29]. FATLBO achieved a solution superior to all other methods.

4.2.2 Tubular column

A uniform column of tubular section is designed with hinge joints installed at both ends (see Fig. 8). The column must carry a compressive load $P = 2500$ kgf at the lowest cost [30]. The column is made of a material with a yield strength (σ_y) of 500 kgf/cm^2, modulus of elasticity (E) of 0.85×10^6 kgf/cm^2, and weight density (ρ) of 0.0025 kgf/cm^3. Column length (L) is 250 cm. Column stress should

Table 4 Comparative results of FATLBO with GA, DE, PSO, BA, and PBA

No	Functions		Min	GA	DE	PSO	BA	PBA	TLBO	FATLBO
1	Beale	Mean	0	*0*	*0*	*0*	1.88E−05	*0*	*0*	*0*
		StdDev		*0*	*0*	*0*	1.94E−05	*0*	*0*	*0*
2	Easom	Mean	−1	*−1*	*−1*	*−1*	−0.99994	*−1*	*−1*	*−1*
		StdDev		*0*	*0*	*0*	4.50E−05	*0*	*0*	*0*
3	Matyas	Mean	0	*0*	*0*	*0*	*0*	*0*	*0*	*0*
		StdDev		*0*	*0*	*0*	*0*	*0*	*0*	*0*
4	Bohachevsky 1	Mean	0	*0*	*0*	*0*	*0*	*0*	*0*	*0*
		StdDev		*0*	*0*	*0*	*0*	*0*	*0*	*0*
5	Booth	Mean	0	*0*	*0*	*0*	0.00053	*0*	*0*	*0*
		StdDev		*0*	*0*	*0*	0.00074	*0*	*0*	*0*
6	Michalewicz 2	Mean	−1.8013	*−1.8013*	*−1.8013*	−1.57287	*−1.8013*	*−1.8013*	*−1.8013*	*−1.8013*
		StdDev		*0*	*0*	0.11986	*0*	*0*	*0*	*0*
7	Schaffer	Mean	0	0.00424	*0*	*0*	*0*	*0*	*0*	*0*
		StdDev		0.00476	*0*	*0*	*0*	*0*	*0*	*0*
8	Six Hump Camel Back	Mean	−1.03163	*−1.03163*	*−1.03163*	*−1.03163*	*−1.03163*	*−1.03163*	*−1.03163*	*−1.03163*
		StdDev		*0*	*0*	*0*	*0*	*0*	*0*	*0*
9	Boachevsky 2	Mean	0	0.06829	*0*	*0*	*0*	*0*	*0*	*0*
		StdDev		0.07822	*0*	*0*	*0*	*0*	*0*	*0*
10	Boachevsky 3	Mean	0	*0*	*0*	*0*	*0*	*0*	*0*	*0*
		StdDev		*0*	*0*	*0*	*0*	*0*	*0*	*0*
11	Shubert	Mean	−186.73	*−186.73*	*−186.73*	*−186.73*	*−186.73*	*−186.73*	*−186.73*	*−186.73*
		StdDev		*0*	*0*	*0*	*0*	*0*	*0*	*0*
12	Colville	Mean	0	0.01494	0.04091	*0*	1.11760	*0*	*0*	*0*
		StdDev		0.00736	0.08198	*0*	0.46623	*0*	*0*	*0*
13	Michalewicz 5	Mean	−4.6877	−4.64483	−4.68348	−2.49087	*−4.6877*	*−4.6877*	*−4.6877*	*−4.6877*
		StdDev		0.09785	0.01253	0.25695	*0*	*0*	*0*	*0*
14	Zakharov	Mean	0	0.01336	*0*	*0*	*0*	*0*	*0*	*0*
		StdDev		0.00453	*0*	*0*	*0*	*0*	*0*	*0*
15	Michalewicz10	Mean	−9.6602	−9.49683	−9.59115	−4.00718	*−9.6602*	*−9.6602*	−9.6172	*−9.6602*
		StdDev		0.14112	0.06421	0.50263	*0*	*0*	4.52E−02	*0*
16	Step	Mean	0	1.17E+03	*0*	*0*	5.12370	*0*	*0*	*0*
		StdDev		76.56145	*0*	*0*	0.39209	*0*	*0*	*0*
17	Sphere	Mean	0	1.11E+03	*0*	*0*	*0*	*0*	*0*	*0*
		StdDev		74.21447	*0*	*0*	*0*	*0*	*0*	*0*
18	Sum Squares	Mean	0	1.48E+02	*0*	*0*	*0*	*0*	*0*	*0*
		StdDev		12.40929	*0*	*0*	*0*	*0*	*0*	*0*
19	Quartic	Mean	0	0.18070	0.00136	0.00116	1.72E−06	0.00678	7.49E−03	3.52E−04
		StdDev		0.02712	0.00042	0.00028	1.85E−06	0.00133	1.99E−03	1.61E−04
20	Schwefel 2.22	Mean	0	11.0214	*0*	*0*	*0*	7.59E−10	*0*	*0*
		StdDev		1.38686	*0*	*0*	*0*	7.10E−10	*0*	*0*
21	Schwefel 1.2	Mean	0	7.40E+03	*0*	*0*	*0*	*0*	*0*	*0*
		StdDev		1.14E+03	*0*	*0*	*0*	*0*	*0*	*0*
22	Rosenbrock	Mean	0	1.96E+05	18.20394	15.088617	28.834	4.2831	1.04E−07	3.80E−07
		StdDev		3.85E+04	5.03619	24.170196	0.10597	5.7877	2.95E−07	4.84E−07
23	Dixon-Price	Mean	0	1.22E+03	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667
		StdDev		2.66E+02	E−9	E−8	1.16E−09	5.65E−10	0	3.34E−15
24	Rastrigin	Mean	0	52.92259	11.71673	43.97714	*0*	*0*	*0*	*0*
		StdDev		4.56486	2.53817	11.72868	*0*	*0*	*0*	*0*

Table 4 continued

No	Functions		Min	GA	DE	PSO	BA	PBA	TLBO	FATLBO
25	Griewank	Mean	0	10.63346	0.00148	0.01739	*0*	0.00468	*0*	*0*
		StdDev		1.16146	0.00296	0.02081	*0*	0.00672	*0*	*0*
26	Ackley	Mean	0	14.67178	*0*	0.16462	*0*	3.12E−08	*0*	*0*
		StdDev		0.17814	*0*	0.49387	*0*	3.98E−08	*0*	*0*
Count of algorithm found global minimum				9	18	17	18	20	*21*	*23*

The number in italics indicates the best value

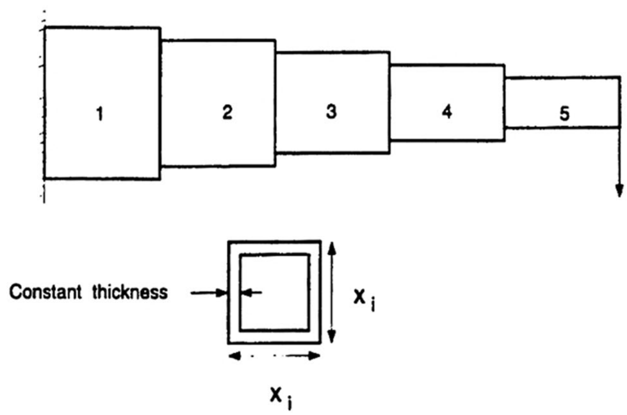


Fig. 7 Cantilever beam problem

be less than the buckling stress (constraint g_1) and yield stress (constraint g_2). Mean column diameter is restricted to between 2 and 14 cm, and columns with thicknesses outside the range 0.2–0.8 cm are not available in the market. The cost of the column includes material and construction costs. The cost is the objective function and expressed as $5W + 2d$. W is the weight in kilograms-force and d is the mean diameter of the column in centimeters.

The optimization model for this problem is given as follows:

Minimize:

$$f(d, t) = 9.8dt + 2d \tag{7}$$

Subject to:

$$g_1 = \frac{P}{\pi dt} \leq \sigma_y \tag{8}$$

$$g_2 = \frac{P}{\pi dt} - \frac{\pi^2 E(d^2 + t^2)}{8L^2} \leq 0 \tag{9}$$

$$g_3 = -d + 2 \leq 0 \tag{10}$$

$$g_4 = d - 14 \leq 0 \tag{11}$$

$$g_5 = -t + 0.2 \leq 0 \tag{12}$$

$$g_6 = t - 0.8 \leq 0 \tag{13}$$

Table 5 compares FATLBO-obtained results with those of other methods reported previously in the literature [29–31]. Table 6 shows that, although other methods obtained the best objective value, they did so in violation of constraint g_2. FATLBO, however, generated the best result within all constraints provided. FATLBO also identified the best result using a significantly lower number of evaluations than CS.

Table 5 Best solution for the cantilever beam design

	Chickermane and Gea				Gandomi et al.	Present Study
	CONLIN	MMA	GCA(I)	GCA(II)	CS	FATLBO
x_1	6.0100	6.0100	6.0100	6.0100	6.0089	6.01601
x_2	5.3000	5.3000	5.3000	5.3000	5.3049	5.30917
x_3	4.4900	4.4900	4.4900	4.4900	4.5023	4.49435
x_4	3.4900	3.4900	3.4900	3.4900	3.5077	3.50147
x_5	2.1500	2.1500	2.1500	2.1500	2.1504	2.15267
f_{min}	N.C.[a]	1.3400	1.3400	1.3400	1.33999	1.33996
Average	N.A.	N.A.	N.A.	N.A.	N.A.	1.33996
Standard deviation	N.A.	N.A.	N.A.	N.A.	N.A.	4.56E−5
No. evaluations	N.A.	N.A.	N.A.	N.A.	N.A.	15,000

[a] Not converge

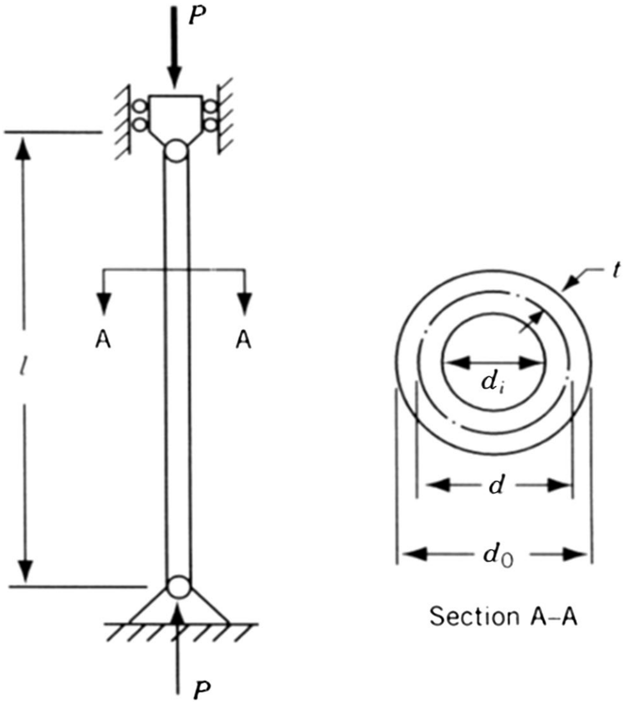


Fig. 8 Tubular column problem

4.2.3 *Three-bar truss design*

This case considers a 3-bar truss design problem [32], as shown in Fig. 9. The objective of this case is to minimize the volume of a statistically loaded 3-bar truss subject to stress (*s*) constraints. The problem involves two decision variables: cross-sectional areas (*A*1, *A*2). The problem may be stated as:

Minimize:

$$f(A_1, A_2) = (2\sqrt{2}A_1 + A_2) \times l \tag{14}$$

Subject to:

$$g_1 = \frac{\sqrt{2}A_1 + A_2}{\sqrt{2}A_1^2 + 2A_1A_2}P \le \sigma \tag{15}$$

$$g_2 = \frac{A_2}{\sqrt{2}A_1^2 + 2A_1A_2}P \le \sigma \tag{16}$$

$$g_3 = \frac{1}{A_1 + \sqrt{2}A_2}P \le \sigma \tag{17}$$

where $0 \le A_1, A_2 \le 1$; $l = 100$ cm, $P = 2$ KN/cm^2, and $\sigma = 2$ KN/cm^2.

This design is a nonlinear fractional programming problem. Table 7 represents solutions obtained by several researchers [29, 32, 33]. The solution obtained by FATLBO is $(A_1, A_2) = (0.78868, 0.40825)$ with an objective value equal to 263.8958. Although Tsai reported the best objective value, the result is not feasible due to its violation of the first constraint. Results obtained by FATLBO are thus superior to those of previous studies. FATLBO also is the most efficient method to solve this problem, as it required only 6000 evaluation cycles compared with the 15,000 required by CS.

4.2.4 *Reinforced concrete beam structure*

Amir and Hasegawa presented a simplified optimization of total reinforced concrete beam cost, shown in Fig. 10 [34]. The beam is assumed simply supported with a span of 30 ft and subjected to a live load of 2.0 klbf and a dead load

Table 6 Best solution for the tubular column example

	Hsu and Liu Fuzzy PD controller	Rao Mathematical programming	Gandomi et al. CS	Present study FATLBO
d (cm)	5.5407	5.44	5.45139	5.45116
t (cm)	0.292	0.293	0.29196	0.29197
g_1	499.9827	499.2564	499.9879	499.9919
g_2	0.0001[a]	0.0026[a]	−0.0001	−0.0001
g_3	−3.4507	−3.44	−3.45139	−3.45116
g_4	−8.5493	−8.56	−8.54861	−8.54224
g_5	−0.092	−0.093	−0.09196	−0.09197
g_6	−0.508	−0.507	−0.50804	−0.50803
f_{min}	26.4991	26.5323	26.53217	26.4995
Average	N.A.	N.A.	26.53504	26.49964
Standard deviation	N.A.	N.A.	1.9E−3	2.2E−4
No. evaluation	N.A.	N.A.	15,000	2500

[a] Represents violated sets

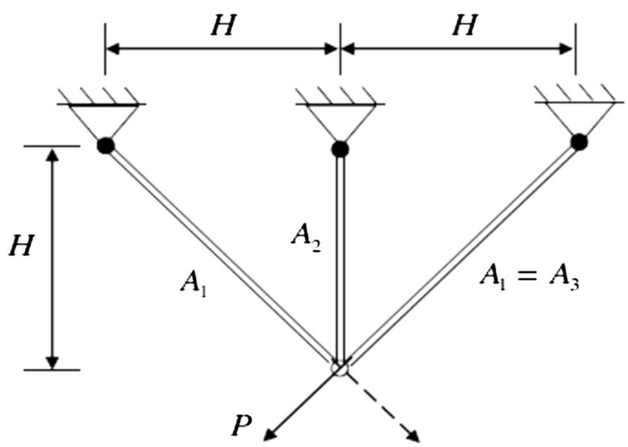


Fig. 9 Three-bar truss problem

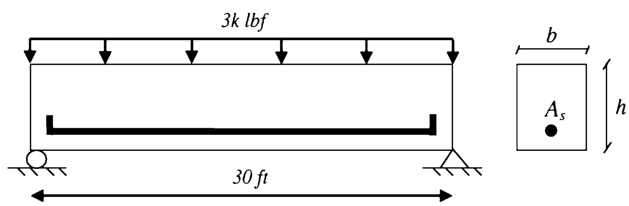


Fig. 10 Reinforced concrete beam problem

of 1.0 klbf, including beam weight. Concrete compressive strength (σ_c) is 5 ksi and the yield stress of the reinforcing steel (σ_y) is 50 ksi. The cost of concrete is 0.02/in^2/linear ft and the cost of steel is \$1.0/in^2/linear ft. Reinforcement area (A_s), beam width (b), and beam depth (h) must be determined to minimize total structure cost. In this paper, the cross-sectional area of the reinforcing bar (A_s) is taken as a discrete type variable that must be chosen from the standard bar dimensions listed in ref [34]; concrete beam width (b) is assumed to be an integer variable; beam depth is a continuous variable; and effective depth is assumed to be 0.8 h.

The structure should be proportioned to attain a strength required under ACI building code 318-77 as follows:

$$M_{\mathrm{u}} = 0.9A_{\mathrm{s}}\sigma_y(0.8)\left(1.0 - 0.59\frac{A_{\mathrm{s}}\sigma_y}{0.8bh\sigma_{\mathrm{c}}}\right) \geq 1.4M_{\mathrm{d}} + 1.7M_{\mathrm{l}} \tag{18}$$

in which M_{u}, M_{d}, and M_{l}, respectively, are the flexural strength, dead load, and live load moments of the beam. In this case, $M_{\mathrm{d}} = 1350$ in kip and $M_{\mathrm{l}} = 2700$ in kip. Beam depth ratio is restricted to be less than or equal to 4. The optimization problem may be stated as:

Minimize:

$$f(A_s, b, h) = 29.4A_{\mathrm{s}} + 0.6bh \tag{19}$$

Subject to:

$$g_1 = \frac{b}{h} \leq 4 \tag{20}$$

$$g_2 = 180 + 7.375\frac{A_{\mathrm{s}}^2}{h} \leq A_{\mathrm{s}}b \tag{21}$$

The variables bound are A_s: [6.0, 6.16, 6.32, 6.6, 7.0, 7.11, 7.2, 7.8, 7.9, 8.0, 8.4] in^2, b: [28, 29, 30, 31 … 38, 39, 40] in, and $5 \leq h \leq 10$ in. This paper also used the constrained functions of $g_1(x)$ and $g_2(x)$ derived by Liebman et al. [35].

Table 8 presents the optimum designs of this problem and parameters used, including several comparisons with prior research on SD-RC [34], GHN-ALM and GHN-EP [36], GA and FLC-AHGA [37], CS [29], FA [38]. In this case study, FATLBO found the same optimum solution identified by FA with a better consistency in mean and standard deviation.

Table 7 Best solution for the three-bar truss design

	Ray and Saini Swarm strategy	Tsai Nonlinear programming	Gandomi et al. CS	Present Study FATLBO
A_1 (cm^2)	0.795	0.788	0.78867	0.78868
A_2 (cm^2)	0.395	0.408	0.40902	0.40852
g_1	1.9966	2.0016[a]	1.9994	2.0000
g_2	0.5191	0.5364	0.5365	0.5361
g_3	1.4775	1.4652	1.4629	1.4637
$f_{\min}$ (cm^3)	264.36	263.68	263.9716	263.8958
Average	N.A.	N.A.	264.0669	263.8959
Standard deviation	N.A.	N.A.	9E−5	3.29E−5
No. evaluation	N.A.	N.A.	15,000	6000

[a] Represents violated sets

Table 8 Result of the reinforced concrete beam example

	Amir and Hasegawa	Shih and Yang		Yun		Gandomi et al.	Gandomi et al.	Present study
	SD-RC	GHN-ALM	GHN-EP	GA	GA-FL	CS	FA	FATLBO
A_s (in^2)	7.8	6.6	6.32	7.2	6.16	6.32	6.32	6.32
b (in)	31	33	34	32	35	34	34	34
h (in)	7.79	8.4952	8.6372	8.0451	8.75	8.5	8.5	8.5
g_1	−0.0205	−0.1155	−0.0635	−0.0224	0	0	0	0
g_2	−4.2012	0.0159	−0.7745	−2.8779	−3.6173	−0.2241	−0.2241	−0.2241
$f_{\min}$ (in^2)	374.2	362.2455	362.0065	366.1459	364.8541	359.2080	359.2080	359.2080
Average	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	460.706	359.7726
Standard deviation	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	80.7387	1.2832
No. evaluation	396	N.A.	N.A.	N.A.	100,000	N.A.	25,000	1000

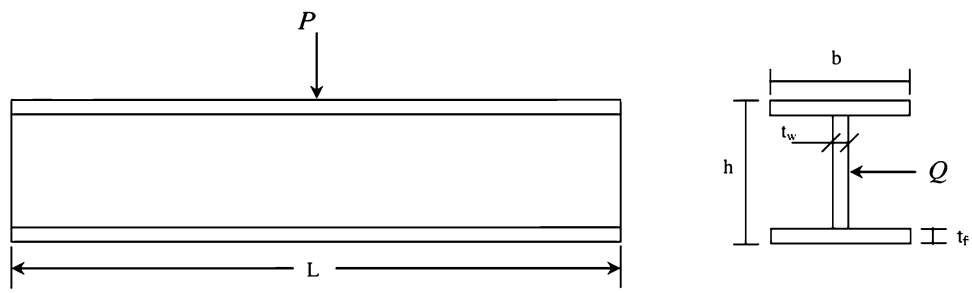


Fig. 11 I-beam design problem

4.2.5 *Minimize I-beam vertical deflection*

This case study, using a design problem with four variables, was modified from an original problem reported in [39]. Figure 11 illustrates the goal of this case-minimizing the vertical deflection of an I-beam. The cross-sectional area and stress constraints must be simultaneously satisfied under given loads.

The objective description of this study is to minimize the vertical deflection $f(x) = \mathrm{PL}^3/48EI$ where beam length (L) and elasticity modulus (E) are, respectively, 5,200 cm and 523,104 kN/cm^2. Thus, the objective function of the problem is considered to be:

Minimize:

$$f(b, h, t_\mathrm{w}, t_\mathrm{f}) = \frac{5000}{\frac{t_\mathrm{w}(h-2t_\mathrm{f})^3}{12} + \frac{bt_\mathrm{f}^3}{6} + 2bt_\mathrm{f}\left(\frac{h-t_\mathrm{f}}{2}\right)^2} \tag{22}$$

Subject to a cross-section area less than 300 cm^2

$$g_1 = 2bt_\mathrm{f} + t_\mathrm{w}(h - 2t_\mathrm{f}) \le 300 \tag{23}$$

If allowable bending stress of the beam is 56 kN/cm^2, the stress constraint is:

$$g_2 = \frac{18h \times 10^4}{t_\mathrm{w}(h-2t_\mathrm{f})^3 + 2bt_\mathrm{w}(4t_\mathrm{f}^2 + 3h(h-2t_\mathrm{f}))} + \frac{15b \times 10^3}{(h-2t_\mathrm{f})t_\mathrm{w}^3 + 2t_\mathrm{w}b^3} \le 56 \tag{24}$$

Table 9 Best solution for I-beam design

	Wang		Gandomi et al.	Present study
	ARSM	Improved ARSM	CS	FATLBO
h (cm)	80.00	79.99	80.000000	80.00000
b (cm)	37.05	48.42	50.000000	50.00000
t_w (cm)	1.71	0.90	0.900000	0.90000
t_f (cm)	2.31	2.40	2.3216715	2.32179
$f_{\min}$ (cm)	0.0157	0.0131	0.0130747	0.0130741
Average	N.A.	N.A.	0.01353646	0.0130884
Standard deviation	N.A.	N.A.	1.3E−4	2.56E−5
No. evaluation	N.A.	N.A.	5000	5000

[a] Represents violated sets

where initial design spaces are $10 \le h \le 80$, $10 \le b \le 50$, $0.9 \le t_\mathrm{w} \le 5$, and $0.9 \le t_\mathrm{f} \le 5$.

For this case study, 5000 completed function evaluations were set as the stopping criterion. Table 9 presents results obtained by FATLBO and other algorithms. This case study has been previously solved using other methods such as adaptive surface method (ARSM), improved ARSM [40], and CS [29]. FATLBO performance surpassed ARSM and improved ARSM in terms of both minimum obtained value and solution average. Although CS matched the results obtained using FATLBO, FATLBO was still slightly better in terms of mean and standard deviation.

5 Conclusion

This paper presents a new optimization algorithm called fuzzy adaptive teaching–learning-based optimization (FATLBO). FATLBO introduces three new modifications to improve the performance of original TLBO namely status monitor, fuzzy adaptive teaching–learning strategies (FATLS), and remedial operator. Its application to sample problems demonstrated the ability of FATLBO to generate solutions at a quality significantly better than other metaheuristic algorithms. Based on mathematical benchmark function results, FATLBO precisely identified 23 of 26 benchmark function solutions, surpassing the performance of GA, DE, BA, PSO, and PBA. FATLBO was also tested with five practical structural design problems. Results demonstrated FATLBO was able to achieve better results with fewer evaluation functions than algorithms tested in previous works. The three new improvements of the FATLBO algorithm are simple to operate. Thus, it can be concluded that the novel FATLBO algorithm, while robust and easy to implement, is able to solve various numerical optimization problem.

This paper also presents a new paradigm to parameter adaptation and can be understood as one of possible ways for fuzzy parameter setting of TLBO. Applying the current fuzzy adaptive paradigm to another new metaheuristic algorithm, e.g., jaya algorithm, is an interesting direction for further research.

References

1. Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. Comput Struct 82(9–10):781–798. doi:10.1016/j.compstruc.2004.01.002
2. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press. Ann Arbor, MI
3. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Paper presented at the proceedings of the IEEE international conference on neural networks, Perth, Australia
4. 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
5. 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
6. Tran D-H, Cheng M-Y, Prayogo D (2016) A novel multiple objective symbiotic organisms search (MOSOS) for time–cost–labor utilization tradeoff problem. Knowl Based Syst 94:132–145. doi:10.1016/j.knosys.2015.11.016
7. 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
8. 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
9. 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
10. 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
11. 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, vol 7077. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp 148–156. doi:10.1007/978-3-642-27242-4_18
12. Naik A, Satapathy SC, Parvathi K (2012) Improvement of initial cluster center of C-means using teaching learning based optimization. Proc Technol 6:428–435. doi:10.1016/j.protcy.2012.10.051
13. Rao RV, Patel V (2013) Multi-objective optimization of two stage thermoelectric cooler using a modified teaching–learning-based optimization algorithm. Eng Appl Artif Intell 26(1):430–445. doi:10.1016/j.engappai.2012.02.016
14. 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
15. Rao RV, Patel V (2013) Multi-objective optimization of heat exchangers using a modified teaching–learning-based optimization algorithm. Appl Math Model 37(3):1147–1162. doi:10.1016/j.apm.2012.03.043
16. Rao RV, Patel V (2011) Multi-objective optimization of combined Brayton and inverse Brayton cycles using advanced optimization algorithms. Eng Optim 44(8):965–983. doi:10.1080/0305215x.2011.624183
17. Niknam T, Kavousi Fard A, 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. doi:10.1016/j.energy.2012.02.023
18. Niknam T, Azizipanah-Abarghooee R, Aghaei J (2013) A new modified teaching-learning algorithm forreserve constrained dynamic economic dispatch. IEEE Trans Power Syst 28(2):749–763. doi:10.1109/TPWRS.2012.2208273
19. Niknam T, Azizipanah-Abarghooee R, Rasoul Narimani M (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. doi:10.1016/j.engappai.2012.07.004
20. Toğan V (2012) Design of planar steel frames using teaching-learning based optimization. Eng Struct 34:225–232. doi:10.1016/j.engstruct.2011.08.035
21. Degertekin SO, Hayalioglu MS (2013) Sizing truss structures using teaching–learning-based optimization. Comput Struct. doi:10.1016/j.compstruc.2012.12.011
22. 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
23. Rao RV (2016) Teaching learning based optimization algorithm and its engineering applications. Springer International Publishing, Switzerland. doi:10.1007/978-3-319-22732-0

24. 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
25. Rao RV, 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
26. Cheng M-Y, Lien L-C (2012) Hybrid artificial intelligence-based PBA for benchmark functions and facility layout design optimization. J Comput Civil Eng 26(5):612–624. doi:10.1061/(ASCE)CP.1943-5487.0000163
27. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Meth Appl Mech Eng 186(2–4):311–338. doi:10.1016/S0045-7825(99)00389-8
28. Chickermane H, Gea HC (1996) Structural optimization using a new local approximation method. Int J Numer Meth Eng 39(5):829–846. doi:10.1002/(sici)1097-0207(19960315)39:5<829:aid-nme884>3.0.co;2-u
29. Gandomi AH, Yang X-S, Alavi AH (2011) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput. doi:10.1007/s00366-011-0241-y
30. Rao SS (1996) Engineering optimization: theory and practice, 3rd edn. Wiley, Chichester
31. Hsu Y-L, Liu T-C (2007) Developing a fuzzy proportional-derivative controller optimization engine for engineering design optimization problems. Eng Optim 39(6):679–700. doi:10.1080/03052150701252664
32. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 33(6):735–748. doi:10.1080/03052150108940941
33. Tsai J-F (2005) Global optimization of nonlinear fractional programming problems in engineering design. Eng Optim 37(4):399–409. doi:10.1080/03052150500066737
34. Amir H, Hasegawa T (1989) Nonlinear mixed-discrete structural optimization. J Struct Eng ASCE 115(3):626–646. doi:10.1061/(asce)0733-9445(1989)115:3(626)
35. Liebman JS, Chanaratna V, Khachaturian N (1981) Discrete structural optimization. J Struct Div ASCE 107(11):2177–2197
36. Shih CJ, Yang YC (2002) Generalized Hopfield network based structural optimization using sequential unconstrained minimization technique with additional penalty strategy. Adv Eng Software 33(7–10):721–729
37. Yun YS (2005) Study on adaptive hybrid genetic algorithm and its applications to engineering design problems. MSc Thesis, Waseda University
38. Gandomi AH, Yang X-S, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. Comput Struct 89(23–24):2325–2336. doi:10.1016/j.compstruc.2011.08.002
39. Gold S, Krishnamurty S (1997) Trade-offs in robust engineering design. In: Paper presented at the proceeding of the 1997 ASME design engineering technical conferences, Sacramento
40. Wang G (2003) Adaptive response surface method using inherited latin hypercube design points. J Mech Des 125(2):210–220. doi:10.1115/1.1561044