Information Sciences 183 (2012) 1–15

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems

R.V. Rao *, V.J. Savsani, D.P. Vakharia

Department of Mechanical Engineering, S.V. National Institute of Technology, Surat 395 007, India

ARTICLE INFO

Article history:
Received 16 November 2010
Received in revised form 6 February 2011
Accepted 9 August 2011
Available online 22 August 2011

Keywords:
Teaching–Learning-Based Optimization
Non-linear optimization problems
Benchmarks functions

ABSTRACT

An efficient optimization method called 'Teaching–Learning-Based Optimization (TLBO)' is proposed in this paper for large scale non-linear optimization problems for finding the global solutions. The proposed method is based on the effect of the influence of a teacher on the output of learners in a class. The basic philosophy of the method is explained in detail. The effectiveness of the method is tested on many benchmark problems with different characteristics and the results are compared with other population based methods.

1. Introduction

Many difficulties such as multimodality, dimensionality and differentiability are associated with the optimization of large scale problems. Traditional techniques like steepest decent, linear programming, dynamic programming, etc. generally fail to solve such large scale problems especially with non-linear objective functions. Most of the traditional techniques require gradient information and hence it is not possible to solve non-differentiable functions with the help of such traditional techniques. Moreover, such techniques often fail to solve optimization problems that have many local optima. So there remains a need for efficient and effective optimization techniques. Continuous research is being conducted in this field and nature-inspired meta-heuristic optimization techniques are proving to be better than the traditional techniques and thus are widely used.

Some of the well known meta-heuristics developed during the last three decades are: Genetic Algorithm (GA) [12] which works on the principle of the Darwinian theory of the survival of the fittest and the theory of evolution of the living beings; Artificial Immune Algorithm (AIA) [9] which works on the immune system of the human being; Ant Colony Optimization (ACO) [6] which works on the foraging behavior of the ant for the food; Particle Swarm Optimization (PSO) [16] which works on the foraging behavior of the swarm of birds; Differential Evolution (DE) [7,26] which is similar to GA with specialized crossover and selection method; Harmony Search (HS) [10] which works on the principle of music improvisation in a music player; Bacteria Foraging Optimization (BFO) [20] which works on the behavior of bacteria; Shuffled Frog Leaping (SFL) [8] which works on the principle of communication among the frogs, Artificial Bee Colony (ABC) [3,13] which works on the foraging behavior of a honey bee; Biogeography-Based Optimization (BBO) [25] which works on the principle of immigration and emigration of the species from one place to the other; Gravitational Search Algorithm (GSA) [22] which works on the principle of gravitational force acting between the bodies; Grenade Explosion Method (GEM) [1] which works on the

* Corresponding author. Tel.: +91 2612201661; fax: +91 2612201571.
E-mail address: ravipudirao@gmail.com (R.V. Rao).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.08.006

principle of explosion of a grenade. These algorithms have been applied to many engineering optimization problems and proved effective to solve some specific kind of problems.

All the above mentioned algorithms are nature-inspired population based optimization methods, but they have some limitations in one or the other aspect. Due to this fact, more research is required to test the algorithms for different problems to check their suitability for the considered problems. Research is continued to enhance the existing algorithms to suit particular applications. Enhancement is done either (a) by modifying the existing algorithms or (b) by hybridizing the existing algorithms. Enhancement due to modifications in the existing algorithms is reported in GA [18], PSO [5,19,27,28,31], ACO [23], ABC [4], HS [11], SFL [17], etc. Enhancement can also be done by combining the strengths of different optimization algorithms, known as hybridization of algorithms. Hybridization is an effective way to make the algorithm efficient and it combines the best properties of different algorithms. Some of such hybridized algorithms can be found in [15,21,29,30,32,33].

The main limitation of all the algorithms mentioned above is that different parameters are required for proper working of these algorithms. Proper selection of the parameters is essential for the searching of the optimum solution by these algorithms. A change in the algorithm parameters changes the effectiveness of the algorithm. Most commonly used evolutionary optimization technique is Genetic Algorithm (GA). However, GA provides a near optimal solution for a complex problem having large number of variables and constraints. This is mainly due to the difficulty in determining the optimum controlling parameters such as population size, crossover rate and mutation rate. The same is the case with PSO, which uses inertia weight, social and cognitive parameters. Similarly, ABC requires optimum controlling parameters of number of bees (employed, scout, and onlookers), limit, etc. HS requires harmony memory consideration rate, pitch adjusting rate, and the number of improvisations. Sometimes, the difficulty for the selection of parameters increases with modifications and hybridization. Therefore, the efforts must be continued to develop an optimization technique which is free from the algorithm parameters, i.e. no algorithm parameters are required for the working of the algorithm. This aspect is considered in the present work.

An optimization method, Teaching–Learning-Based Optimization (TLBO), is proposed in this paper to obtain global solutions for continuous non-linear functions with less computational effort and high consistency. The TLBO method works on the philosophy of teaching and learning. The TLBO method is based on the effect of the influence of a teacher on the output of learners in a class. Here, output is considered in terms of results or grades. The teacher is generally considered as a highly learned person who shares his or her knowledge with the learners. The quality of a teacher affects the outcome of learners. It is obvious that a good teacher trains learners such that they can have better results in terms of their marks or grades. Moreover, learners also learn from interaction between themselves, which also helps in their results.

The detailed explanation of TLBO is given in the next section.

2. Teaching–Learning-Based Optimization

Assume two different teachers, T_1 and T_2, teaching a subject with same content to the same merit level learners in two different classes. Fig. 1 shows the distribution of marks obtained by the learners of two different classes evaluated by the teachers. Curves 1 and 2 represent the marks obtained by the learners taught by teacher T_1 and T_2 respectively. A normal distribution is assumed for the obtained marks, but in actual practice it can have skewness. Normal distribution is defined as,

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

where σ^2 is the variance, μ is the mean and x is any value of which normal distribution function is required.

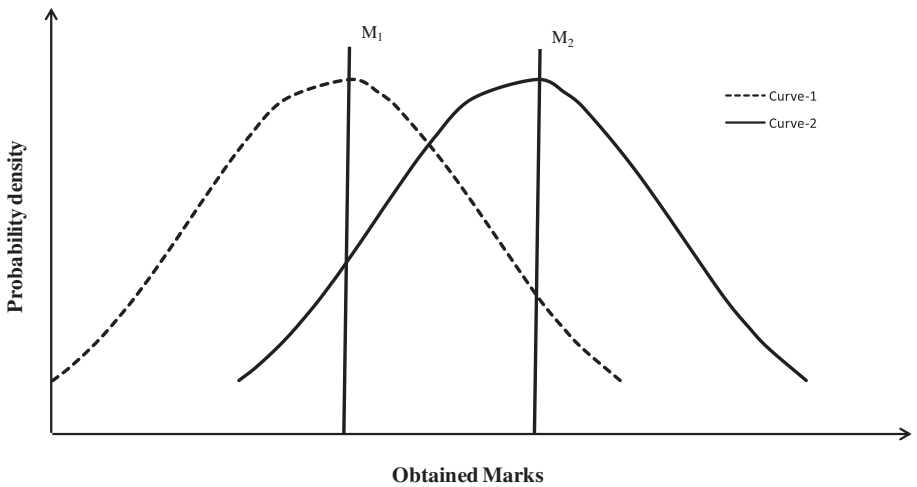

Fig. 1. Distribution of marks obtained by learners taught by two different teachers.

It is seen from Fig. 1 that curve-2 represents better results than curve-1 and so it can be said that teacher T_2 is better than teacher T_1 in terms of teaching. The main difference between both the results is their mean (M_2 for curve-2 and M_1 for curve-1), i.e. a good teacher produces a better mean for the results of the learners. Learners also learn from interaction between themselves, which also helps in their results.

Consider Fig. 2 which shows a model for the marks obtained by the learners in a class with curve-A having mean M_A. Teacher is considered as the most knowledgeable person in the society, so the best learner is mimicked as a teacher, and this is shown by T_A in Fig. 2. The teacher tries to disseminate knowledge among the learners which will in turn increase the knowledge level of the whole class and help learners to get good marks or grades. So a teacher increases the mean of the class according to his or her capability. In Fig. 2, teacher T_A will try to move mean M_A towards their own level according to his or her capability, thereby increasing the learners' level to a new mean M_B. Teacher T_A will put maximum effort into teaching his or her students, but students will gain knowledge according to the quality of teaching delivered by a teacher and the quality of students present in the class. The quality of the students is judged from the mean value of the population. Teacher T_A puts efforts in so as to increase the quality of the students from M_A to M_B, at which stage the students require a new teacher, of superior quality than themselves, i.e. in this case the new teacher is T_B. Hence, there will be a new curve-B with new teacher T_B.

Like other nature-inspired algorithms, TLBO is also a population based method which uses a population of solutions to proceed to the global solution. For TLBO, the population is considered as a group of learners or a class of learners. In optimization algorithms, the population consists of different design variables. In TLBO, different design variables will be analogous to different subjects offered to learners and the learners' result is analogous to the 'fitness', as in other population based optimization techniques. The teacher is considered as the best solution obtained so far.

The process of working of TLBO is divided into two parts. The first part consists of 'Teacher Phase' and the second part consists of 'Learner Phase'. The 'Teacher Phase' means learning from the teacher and the 'Learner Phase' means learning through the interaction between learners.

2.1. Teacher Phase

As shown in Fig. 3, mean of a class increases from M_A to M_B depending upon a good teacher. A good teacher brings his or her learners up to his or her level in terms of knowledge. But in practice this is not possible and a teacher can only move the mean of a class up to some extent depending on the capability of the class. This follows a random process depending on many factors.

Let M_i be the mean and T_i be the teacher at any iteration i. T_i will try to move mean M_i towards its own level, so now the new mean will be T_i designated as M_{new}. The solution is updated according to the difference between the existing and the new mean given by

$$Difference\_Mean_i = r_i(M_{new} - T_F M_i) \tag{2}$$

where T_F is a teaching factor that decides the value of the mean to be changed, and r_i is a random number in the range $[0,1]$. The value of T_F can be either 1 or 2 which is again a heuristic step and decided randomly with equal probability as $T_F = round[1 + rand(0,1)\{2-1\}]$.

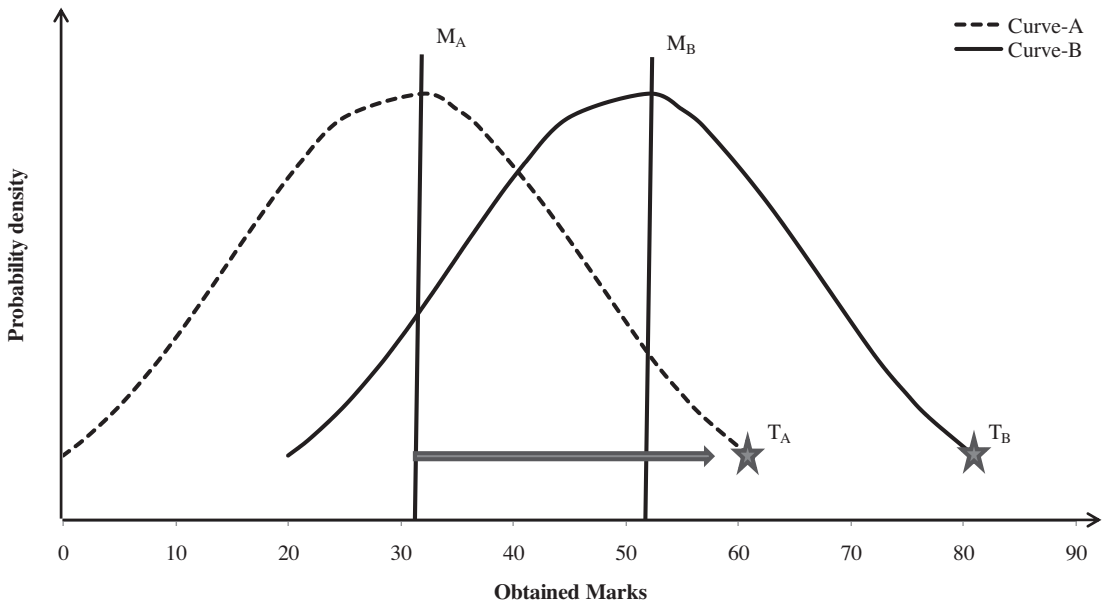

Fig. 2. Model for obtained marks distribution for a group of learners.

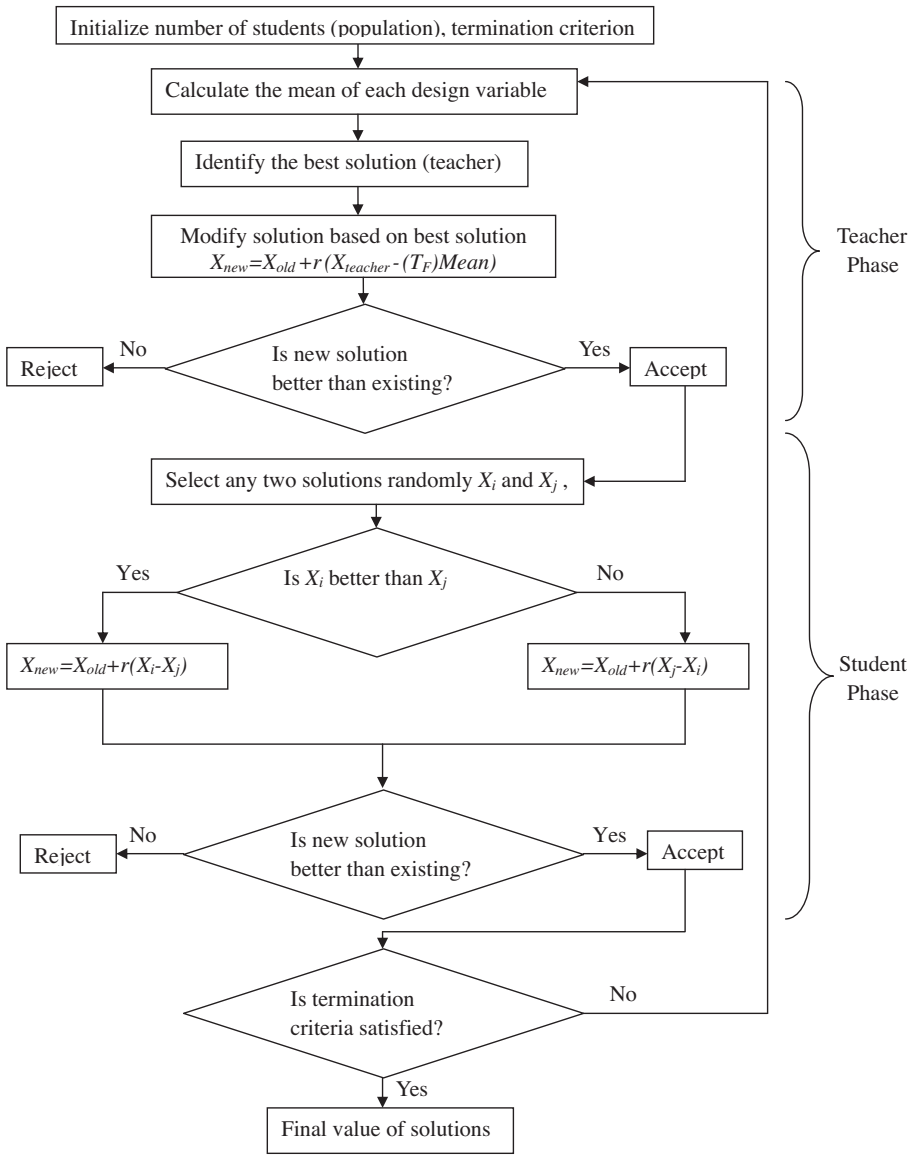

Fig. 3. Flow chart showing the working of TLBO algorithm.

This difference modifies the existing solution according to the following expression

$$X_{new,i} = X_{old,i} + Difference\_Mean_i \tag{3}$$

2.2. Learner Phase

Learners increase their knowledge by two different means: one through input from the teacher and other through interaction between themselves. A learner interacts randomly with other learners with the help of group discussions, presentations, formal communications, etc. A learner learns something new if the other learner has more knowledge than him or her. Learner modification is expressed as,

For $i = 1: P_n$

Randomly select another learner X_j, such that $i \neq j$

If $f(X_i) < f(X_j)$

$X_{new,i} = X_{old,i} + r_i(X_i - X_j)$

Else
$X_{new,i} = X_{old,i} + r_i(X_j - X_i)$
End If
End For
Accept X_{new} if it gives a better function value.
The flow chart for the TLBO method is given in Fig. 3.

3. Implementation of TLBO for optimization

The step-wise procedure for the implementation of TLBO is given in this section. For demonstration of the procedure, Rastrigin function of $[f_(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]]$ is considered. Rastrigin function is a multimodal, separable and regular function. Three dimensional and the contour plots for the Rastrigin function are shown in Fig. 4.

Step 1: Define the optimization problem and initialize the optimization parameters
The following optimization parameters are considered for the Rastrigin function
- Population size $(P_n) = 10$
- Number of generations $(G_n) = 20$
- Number of design variables $(D_n) = 2$
- Limits of design variables $(L_{L,i} \leqslant x_{,i} \leqslant U_{L,I}) = -5.12 \leqslant x_{,1,2} \leqslant 5.12$

Define optimization problem as, Minimize $f(X) =$ Minimize $f_(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$

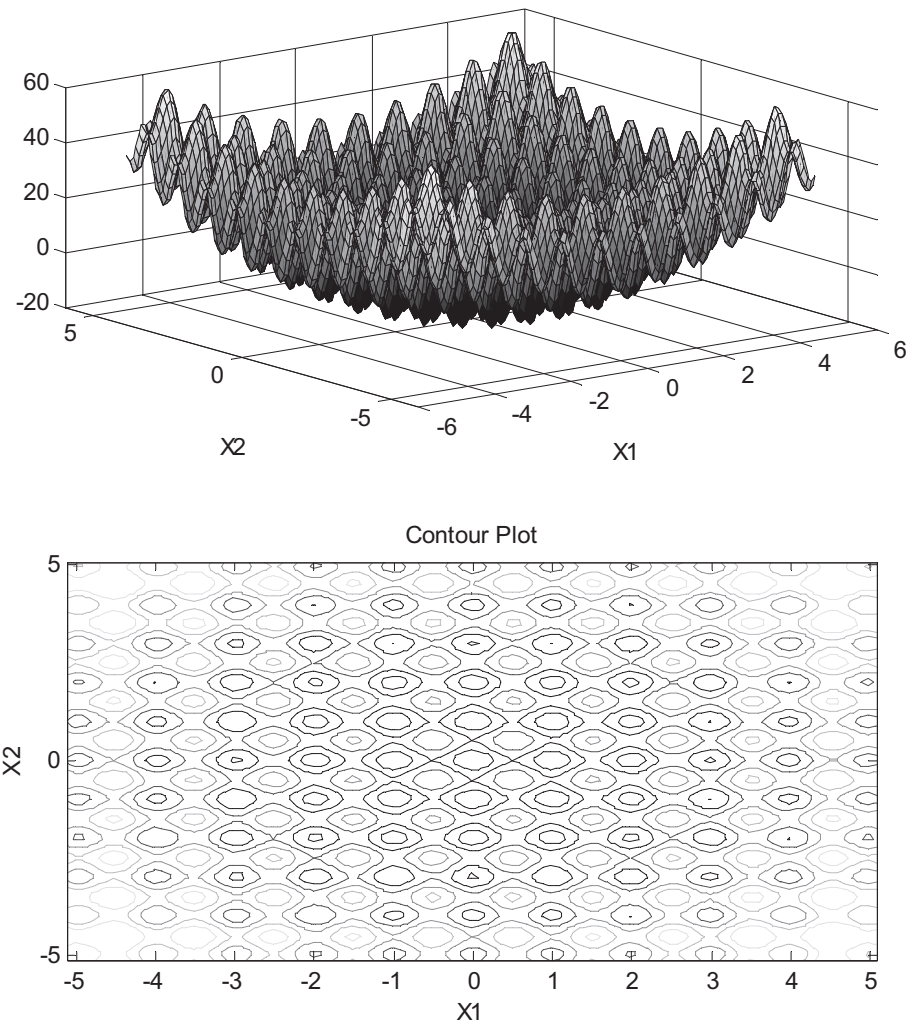

Fig. 4. Three dimensional and contour plots for the Rastrigin function.

Step 2: Initialize the population Generate random population according to the population size and the number of design variables. For TLBO, population size indicates the number of learners and the design variables indicate the subjects (i.e. courses) offered. Short the population according to their respective objective function value. This population is expressed as

$$population = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,D} \\ x_{2,1} & x_{2,2} & \dots & x_{2,D} \\ \vdots & & \vdots & \\ x_{P_n,1} & x_{P_n,2} & \dots & x_{P_n,D} \end{bmatrix} = \begin{bmatrix} 0.7654 & -1.1465 \\ -1.1075 & 2.7974 \\ -1.0483 & 0.6283 \\ -0.9548 & 2.4612 \\ -1.9326 & 3.7264 \\ -4.8123 & -1.0392 \\ -0.3295 & 3.2264 \\ -4.9845 & -1.5195 \\ 4.7380 & -0.4940 \\ -4.6556 & -2.3371 \end{bmatrix}$$ and the corresponding objective function value

$$= \begin{bmatrix} 14.8839 \\ 18.3136 \\ 18.8732 \\ 27.0735 \\ 29.9786 \\ 30.7256 \\ 33.8296 \\ 47.1267 \\ 53.4364 \\ 57.9284 \end{bmatrix}$$

Step 3: Teacher Phase Calculate the mean of the population column wise, which will give the mean for the particular subject as,

$$M_{,D} = [m_1, m_2, \dots, m_D] = [-1.4322, 0.6303] \tag{4}$$

The best solution will act as a teacher for that iteration

$$X_{teacher} = X_{f(X)=\min} = [0.7654, -1.1465] \tag{5}$$

The teacher will try to shift the mean from $M_{,D}$ towards $X_{,teacher}$, which will act as a new mean for the iteration. So,

$$M\_new_{,D} = X_{teacher,D} \tag{6}$$

The difference between two means is expressed as,

$$Difference_{,D} = r(M\_new_{,D} - T_F M_{,D}) \tag{7}$$

The value of T_F is randomly selected as 1 or 2. The obtained difference is added to the current solution to update its values using,

$$X_{new,D} = X_{old,D} + Difference_{,D} = \begin{bmatrix} 3.4434 & -2.5854 \\ 1.5705 & 1.3585 \\ 1.6297 & -0.8106 \\ 1.7232 & 1.0223 \\ 0.7454 & 2.2876 \\ -2.1343 & -2.4781 \\ 2.3485 & 1.7875 \\ -2.3065 & -2.9583 \\ 5.1200 & -1.9329 \\ -1.9776 & -3.7760 \end{bmatrix}$$ and the corresponding objective function value

$$= \begin{bmatrix} 56.5089 \\ 39.6501 \\ 26.4588 \\ 15.7869 \\ 28.4185 \\ 33.9543 \\ 32.1771 \\ 27.8865 \\ 33.5362 \\ 26.6438 \end{bmatrix} \tag{8}$$

Accept X_{new} if it gives better function value.

$$X_{new} = \begin{bmatrix} 0.7654 & -1.1465 \\ -1.1075 & 2.7974 \\ -1.0483 & 0.6283 \\ 1.7232 & 1.0223 \\ 0.7454 & 2.2876 \\ -4.8123 & -1.0392 \\ 2.3485 & 1.7875 \\ -2.3065 & -2.9583 \\ 5.1200 & -1.9329 \\ -1.9776 & -3.7760 \end{bmatrix} \text{ and the corresponding objective function value } = \begin{bmatrix} 14.8839 \\ 18.3136 \\ 18.8732 \\ 15.7869 \\ 28.4185 \\ 30.7256 \\ 32.1771 \\ 27.8865 \\ 33.5362 \\ 26.6438 \end{bmatrix}$$

Step 4: Learner Phase As explained above, learners increase their knowledge with the help of their mutual interaction. The mathematical expression is explained in Section 2.2. Obtain X_{new} after the student phase.

$$X_{new} = \begin{bmatrix} 0.7654 & -1.1465 \\ -1.1479 & 1.1465 \\ 1.7232 & 1.0223 \\ -1.1075 & 2.7974 \\ 0.6497 & 2.7974 \\ -1.0483 & 0.6283 \\ 0.4677 & 1.0451 \\ -1.9397 & -2.7420 \\ -1.6623 & 0.8464 \\ -1.9776 & -3.7760 \end{bmatrix} \text{ and the corresponding objective function value } = \begin{bmatrix} 14.8839 \\ 14.8839 \\ 15.7869 \\ 18.3136 \\ 18.3136 \\ 18.8732 \\ 21.5048 \\ 22.4935 \\ 23.0245 \\ 26.6438 \end{bmatrix}$$

Step 5: Termination criterion Stop if the maximum generation number is achieved; otherwise repeat from step 3.

Table 1
Variation of design variables and objective function (Rastrigin) for the first two generations.

				Teacher Phase			Student Phase		
	X		$f(X)$	X		$f(X)$	X		$f(X)$
	0.7654	**−1.1465**	**14.8839**	**0.7654**	**−1.1465**	**14.8839**	**0.7654**	**−1.1465**	**14.8839**
	−1.1075	2.7974	18.3136	−1.1075	2.7974	18.3136	−1.1479	−1.1465	14.8839
	−1.0483	0.6283	18.8732	−1.0483	0.6283	18.8732	1.7232	1.0223	15.7869
Generation-1	−0.9548	2.4612	27.0735	1.7232	1.0223	15.7869	−1.1075	2.7974	18.3136
	−1.9326	3.7264	29.9786	0.7454	2.2876	28.4185	0.6497	2.7974	18.3136
	−4.8123	−1.0392	30.7256	−4.8123	−1.0392	30.7256	−1.0483	0.6283	18.8732
	−0.3295	3.2264	33.8296	2.3485	1.7875	32.1771	0.4677	1.0451	21.5048
	−4.9845	−1.5195	47.1267	−2.3065	−2.9583	27.8865	−1.9397	−2.7420	22.4935
	4.7380	−0.4940	53.4364	5.1200	−1.9329	33.5362	−1.6623	0.8464	23.0245
	−4.6556	−2.3371	57.9284	−1.9776	−3.7760	26.6438	−1.9776	−3.7760	26.6438
Mean	−1.4322	0.6303							
$TF = 2$									
$F(X)$ average			33.21695			24.72453			19.47217
	0.7654	**−1.1465**	**14.8839**	**0.7654**	**−1.1465**	**14.8839**	**0.0794**	**0.0270**	**1.3684**
	−1.1479	−1.1465	14.8839	−0.0202	−1.7478	13.2735	1.9560	1.0181	5.3067
	1.7232	1.0223	15.7869	1.7232	1.0223	15.7869	0.0579	2.0991	6.9402
Generation-2	−1.1075	2.7974	18.3136	0.0202	2.1961	11.5833	0.0395	2.1092	7.0185
	0.6497	2.7974	18.3136	0.6497	2.7974	18.3136	0.0202	2.1961	11.5833
	−1.0483	0.6283	18.8732	0.0794	0.0270	1.3684	−0.0202	−1.7478	13.2735
	0.4677	1.0451	21.5048	0.4677	1.0451	21.5048	0.6497	2.7974	18.3136
	−1.9397	−2.7420	22.4935	−1.9397	−2.7420	22.4935	−1.9397	−2.7420	22.4935
	−1.6623	0.8464	23.0245	−1.6623	0.8464	23.0245	−1.6623	0.8464	23.0245
	−1.9776	−3.7760	26.6438	−1.9776	−3.7760	26.6438	−1.9776	−3.7760	26.6438
Mean	−0.5277	0.0326							
$TF = 1$									
F average			19.47217			16.88762			13.5966

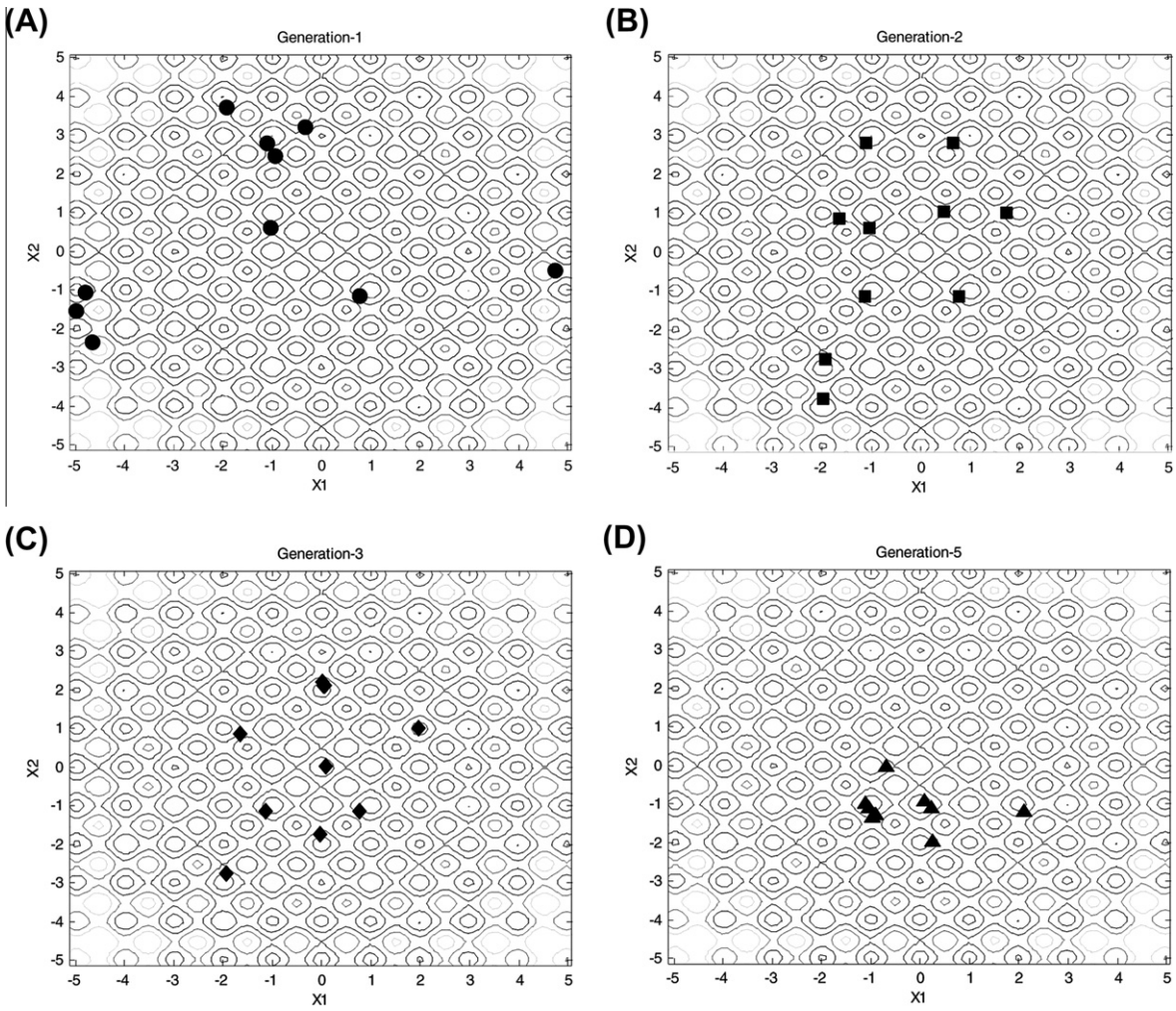

Fig. 5. Visualization of convergence of solutions for Rastigin function, (A) generation-1, (B) generation-2, (C) generation-3, (D) generation-5, (E) generation-10, and (F) generation-20.

Detailed variation of the design variables and the objective function is given in Table 1 for two generations. It is observed from the table that the average (*F(X) average*) and the best value (given in bold) of the objective function decreases as the algorithm proceeds from teacher phase to the student phase in the same generation and decreases with the number of generations. Hence, it can be concluded that the algorithm guarantees convergence. The visualization for the convergence is presented in Fig. 5 for generation numbers of 1, 2, 3, 5, 10, and 20.

4. Comparison of TLBO with other optimization techniques

Like GA, PSO, ABC, HS, etc., TLBO is also a population based technique which implements a group of solutions to proceed for the optimum solution. Many optimization methods require algorithm parameters that affect the performance of the algorithm. GA requires crossover probability, mutation rate, and selection method; PSO requires learning factors, variation of weight, and maximum value of velocity; ABC requires number of employed bees, onlooker bees and value of limit; HS requires harmony memory consideration rate, pitch adjusting rate, and number of improvisations: SFLA requires number of memeplexes, iteration per memeplexes: ACO requires exponent parameters, pheromone evaporation rate and reward factor. Unlike other optimization techniques TLBO does not require any algorithm parameters to be tuned, thus making the implementation of TLBO simpler. As in PSO, TLBO uses the best solution of the iteration to change the existing solution in the population thereby increasing the convergence rate. TLBO does not divide the population like ABC and SFLA. Like GA which uses selection, crossover and mutation phase and ABC which uses employed, onlooker and scout bees phase, TLBO uses two different phases, 'Teacher Phase' and 'Learner Phase'. TLBO uses the mean value of the population to update the solution. TLBO implements greediness to accept the good solution like ABC.

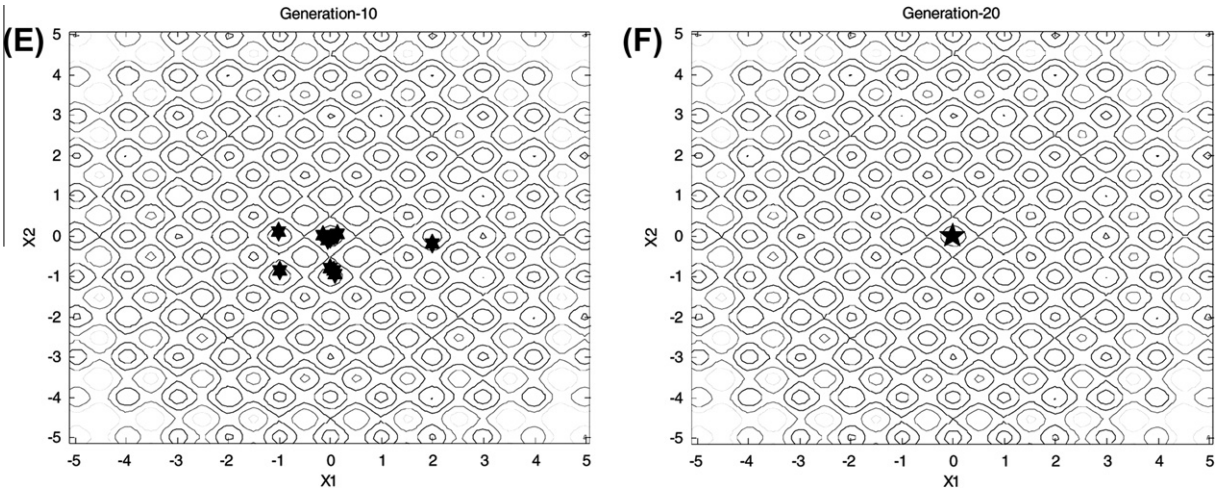

Fig. 5 (*continued*)

The strength of TLBO method is that it does not require any parameters setting for the working of the algorithm. Future research will consist of checking the TLBO method for real life problems and for more challenging benchmark problems. Some improvements will also be incorporated in the TLBO algorithm and the improved method will be checked for benchmark and real life problems.

5. Experimental studies

In the field of optimization it is a common practice to compare different algorithms using different benchmark problems. In this work also different benchmark problems are considered having different characteristics such as separability, multimodality, and regularity. A function is multimodal if it has two or more local optima. A function is separable if it can be written as a sum of functions of variables separately. A function is regular if it is differentiable at each point of its domain. Non-separable functions are more difficult to optimize and difficulty increases if the function is multimodal. Complexity increases when the local optima are randomly distributed. Moreover, complexity increases with the increase in the dimensionality. To validate the proposed algorithm, its results are compared with the results of different algorithms for different benchmark problems available in the literatures. Details of benchmark functions are given in the Appendix.

5.1. Experiment 1

In this experiment, 5 different benchmark problems as given in Ahrari and Atai [1] are optimized using the proposed method and its results are compared with the results of well known optimization techniques like GA, Ant colony system (ANTS), Bee Algorithm (BA) and Grenade Explosion Method (GEM). Description of the benchmark problems is given in Table 2.

Two different criteria are taken for the comparison of algorithms, viz. success percentage and mean function evaluations required. Success percentage indicates the consistency of the algorithm to find the results in different runs and mean function evaluations required indicates the computational effort of the algorithm. In this experiment, an algorithm is considered as successful if the difference between the best value obtained by the algorithm and the global optimum value is less than 0.1% of global optimum or less than 0.001, whichever is smaller [1].

An algorithm is tested for 100 independent runs with the population size of 20 (except for function # 5 for which population size is taken as 50) and maximum number of generations of 50. Average of the function evaluations required and the

Table 2
Details of benchmark functions considered by Arhari and Atai [1].

S.No.	Function	Interval	Global optimum
1	De Jong	$[-2.048, 2.048]$	$F_{min} = 3905.93$, $X = (1,1)$
2	Goldstein and Price	$[-2,2]$	$F_{min} = 3$, $X = (1,1)$
3	Martin and Gaddy	$[0,10]$	$F_{min} = 0$, $X = (5,5)$
4	Rosenbrock ($D = 1$)	(a) $[-1.2, 1.2]$	
		(b) $[-10, 10]$	$F_{min} = 0$, $X = (1,1)$
5	Rosenbrock ($D = 3$)	$[-1.2, 1.2]$	$F_{min} = 0$, $X = (1,1,1,1)$
6	Hyper Sphere ($D = 6$)	$[-5.12, 5.12]$	$F_{min} = 0$, $X = (0,0,0,0,0,0)$

Table 3
Comparison of results for the success percentage and mean number of function evaluations for GA, ANTS, Bee Colony, GEM and TLBO.

	GA		ANTS		Bee Colony		GEM		TLBO	
	Succ. %	Mean no. of FE	Succ. %	Mean no. of FE	Succ. %	Mean no. of FE	Succ. %	Mean no. of FE	Succ. %	Mean no. of FE
1	100	10160	100	6000	100	868	100	746	100	676
2	100	5662	100	5330	100	999	100	701	100	649
3	100	2488	100	1688	100	526	100	258	100	243
4a	100	10212	100	6842	100	631	100	572	100	541
4b	—	—	100	7505	100	2306	100	2289	100	1082
5	—	—	100	8471	100	28529	100	82188	100	2563
6	100	15468	100	22050	100	7113	100	423	100	308

Table 4
Details of benchmark functions considered by Shu and Erwie [24].

S. No.	Function	Interval	Global optimum
1	Powell badly scaled function	[−50,50]	$F_{min} = 0$, $X = (1.098..e{-}5, 9.106..)$
2	B2 function	[−50,50]	$F_{min} = 0$, $X = (0,0)$
3	Booth function	[−50,50]	$F_{min} = 0$, $X = (1,3)$
4	Griewank ($D = 10$)	[−50,50]	$F_{min} = 0$, $X = (0,0,\ldots)$
5	Rastrigin ($D = 10$)	[−50,50]	$F_{min} = 0$, $X = (0,0)$
6	Sphere ($D = 30$)	[−50,50]	$F_{min} = 0$, $X = (0,0,\ldots)$
7	Griewank ($D = 50$)	[−50,50]	$F_{min} = 0$, $X = (0,0,\ldots)$

Table 5
Comparison of results for the success percentage, mean number of function evaluations and the error for PSO, NM-PSO and TLBO.

	PSO			NM-PSO			TLBO		
	Succ. %	Mean no. of FE	Error	Succ. %	Mean no. of FE	Error	Succ. %	Mean no. of FE	Error
1	94	20242	9.89E−06	100	2971	3.78E−06	100	2867	4.0036E−06
2	100	4188	1.4E−08	100	1124	3.23E−10	100	1048	0
J	100	3848	2.6E−08	100	1065	1.26E−09	100	654	3.8293E−12
4	0	(504657)		82	14076	1.04E−11	100	1059	0
5	30	510050	1.08E−04	60	12353	1.91E−11	100	1134	0
6	0	(4530150)		100	87004	2.76E−11	100	1543	0
7	0	(2550250)		82	378354	9.96E−12	100	1857	0

Table 6
Details of benchmark functions considered by Karaboga and Akay [14].

S.No.	Function	Interval	Global optimum
1	Sphere	[−100,100]	$F_{min} = 0$, $X = (0,0,\ldots)$
2	Rosenbrock	[−30,30]	$F_{min} = 0$, $X = (1,1,\ldots)$
3	Rastrigin	[−5.12,5.12]	$F_{min} = 0$, $X = (0,0\ldots)$
4	Griewank	[−600,600]	$F_{min} = 0$, $X = (0,0\ldots)$
5	Ackley	[−32,32]	$F_{min} = 0$, $X = (0,0\ldots)$

success percentage are presented in Table 3. Results of the other algorithms except TLBO are taken from Ahrari and Atai [1]. From Table 3, it can be seen that for all the considered benchmark functions, TLBO requires less number of mean function evaluations with very high consistency of 100% success. The results for the functions 1 to 4a and 6 are better but nearer to the results of GEM given by Ahrari and Atai [1]. The TLBO method has shown much better results for functions 4b and 5 (twice better than those for function 4b and 40 times better than those for function 5) when compared to the results given by Ahrari and Atai [1]. This experiment shows that TLBO is effective in terms of the computational effort and the consistency.

5.2. Experiment 2

In this experiment, 6 different benchmark functions are considered and the results are compared with those given by Particle Swarm Optimization (PSO) and a hybrid Nelder Mead simplex search with PSO called NM-PSO [24]. Details of the problems taken from Shu and Erwie [24] are shown in Table 4.

Table 7
Comparison of results for the mean and the standard deviation for HS, IBA, ABC and TLBO.

		HS		IBA		ABC		TLBO	
	D	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Sphere	5	3.20E−10	2.89E−10	3.91E−17	1.24E−17	4.30E−17	1.07E−17	5.03E−33	1.27E−32
	10	6.45E−08	3.07E−08	4.95E−17	2.30E−17	7.36E−17	4.43E−17	2.26E−29	3.61E−29
	30	7.21E+00	3.62E+00	2.92E−16	6.77E−17	4.69E−16	1.07E−16	6.90E−26	3.18E−25
	50	5.46E+02	9.27E+01	5.39E−16	1.07E−16	1.19E−15	4.68E−16	8.71E−26	1.86E−25
	100	1.90E+04	1.78E+03	1.45E−15	1.63E−16	1.99E−06	2.26E−06	9.42E−26	3.70E−25
Rosenbrock	5	5.94E+00	6.71E+00	4.55E−01	1.54E+00	2.33E−01	2.24E−01	1.80E−01	8.04E−02
	10	6.52E+00	8.16E+00	1.10E+01	2.55E+01	4.62E−01	5.44E−01	5.58E+00	6.18E−01
	30	3.82E+02	5.29E+02	7.57E+01	1.16E+02	9.98E−01	1.52E+00	2.71E+01	1.14E+00
	50	2.47E+04	1.02E+04	6.30E+02	1.20E+03	4.33E+00	5.48E+00	4.78E+01	1.01E+00
	100	1.45E+07	2.16E+06	6.42E+02	8.20E+02	1.12E+02	6.92E+01	9.81E+01	3.61E−01
Ackley	5	2.68E−05	1.24E−05	6.35E−10	9.77E−11	9.64E−17	5.24E−17	0.00E+00	0.00E+00
	10	2.76E−04	7.58E−05	6.71E−02	3.61E−01	3.51E−16	6.13E−17	0.00E+00	0.00E+00
	30	9.43E−01	5.63E−01	1.75E+00	9.32E−01	3.86E−15	3.16E−15	7.11E−16	1.82E−15
	50	5.28E+00	4.03E−01	8.43E+00	7.70E+00	4.38E−08	4.65E−08	1.24E−15	1.95E−15
	100	1.32E+01	4.90E−01	1.89E+01	8.50E−01	1.32E−02	1.30E−02	2.13E−15	1.19E−15
Griewank	5	2.60E−02	1.38E−02	3.14E+00	1.41E+00	4.04E−17	1.12E−17	0.00E+00	0.00E+00
	10	0.00E+00	3.02E−02	1.04E+00	1.13E+00	6.96E−17	4.06E−17	0.00E+00	0.00E+00
	30	1.09E+00	3.92E−02	6.68E+00	6.43E+00	5.82E−06	3.13E−05	0.00E+00	0.00E+00
	50	5.81E+00	9.16E−01	1.34E+02	2.41E+01	5.72E−01	9.22E−01	0.00E+00	0.00E+00
	100	1.78E+02	1.98E+01	7.93E+02	7.96E+01	1.31E+01	6.30E+00	0.00E+00	0.00E+00
Rastrigin	5	6.07E−08	5.52E−08	4.58E+00	2.31E+00	4.34E−17	110E−17	0.00E+00	0.00E+00
	10	1.05E−05	5.23E−06	2.20E+01	7.46E+00	5.77E−17	2.98E−17	0.00E+00	0.00E+00
	30	7.40E−01	7.00E−01	1.28E+02	2.49E+01	4.80E−05	243E−04	0.00E+00	0.00E+00
	50	3.76E+01	4.87E+00	2.72E+02	3.27E+01	4.72E−01	4.92E−01	0.00E+00	0.00E+00
	100	3.15E+02	2.33E+01	6.49E+02	4.52E+01	1.46E+01	4.18E+00	0.00E+00	0.00E+00

Table 8
Details of benchmark functions considered by Akay and Karaboga [2].

Name	Function	Multimodal?	Separable?	Regular?
Sphere	$-100 \leqslant x_i, \leqslant 100$	No	Yes	Yes
Rosenbrock	$-100 \leqslant x_i, \leqslant 100$	No	No	Yes
Step	$-100 \leqslant x_i, \leqslant 100$	No	Yes	No
Schwefel	$-500 \leqslant x_i, \leqslant 500$	Yes	Yes	No
Rastrigin	$-5.12 \leqslant x_i, \leqslant 5.12$	Yes	Yes	Yes
Ackley	$-32 \leqslant x_i, \leqslant 32$	Yes	No	Yes
Griewank	$-600 \leqslant x_i, \leqslant 600$	Yes	No	Yes
Penalised	$-50 \leqslant x_i, \leqslant 50$	Yes	No	Yes

Table 9
Comparison of results for the mean solution for PSO, DE, ABC and TLBO.

	PSO	DE	ABC	TLBO
Sphere	181.16	20.33	8.71E−07	2.02073E−52
Step	1621	1998.03	0	0
Schwefel	−98168.1	−138152.03	−190906.66	−184297.381
Rosenbrock	1.09E+06	8.72E+10	1007.87	497.91
Rastrigin	1033.04	594.69	87.96	0
Griewank	2.2	0.645	0	0
Ackley	3.69	13	0.058	0
Penalized	5.29	1.48E+10	3.46E−08	0.06292

Three different criteria are considered in this experiment viz. success percentage, mean function evaluations required and the error. Error is the average difference between the obtained best solution and the global solution, which indicates the ability of the algorithm to reach the global optimum solution. The algorithm is considered successful if the difference between the obtained best solution and the global optimum is less than 0.001 [24]. The mean function evaluations are obtained only for the successful runs.

For this experiment, the results are obtained for 100 independent runs with population size of 20 and maximum number of generations of 200. The results other than TLBO are taken from Shu and Erwie [24] and are presented in Table 5. It can be

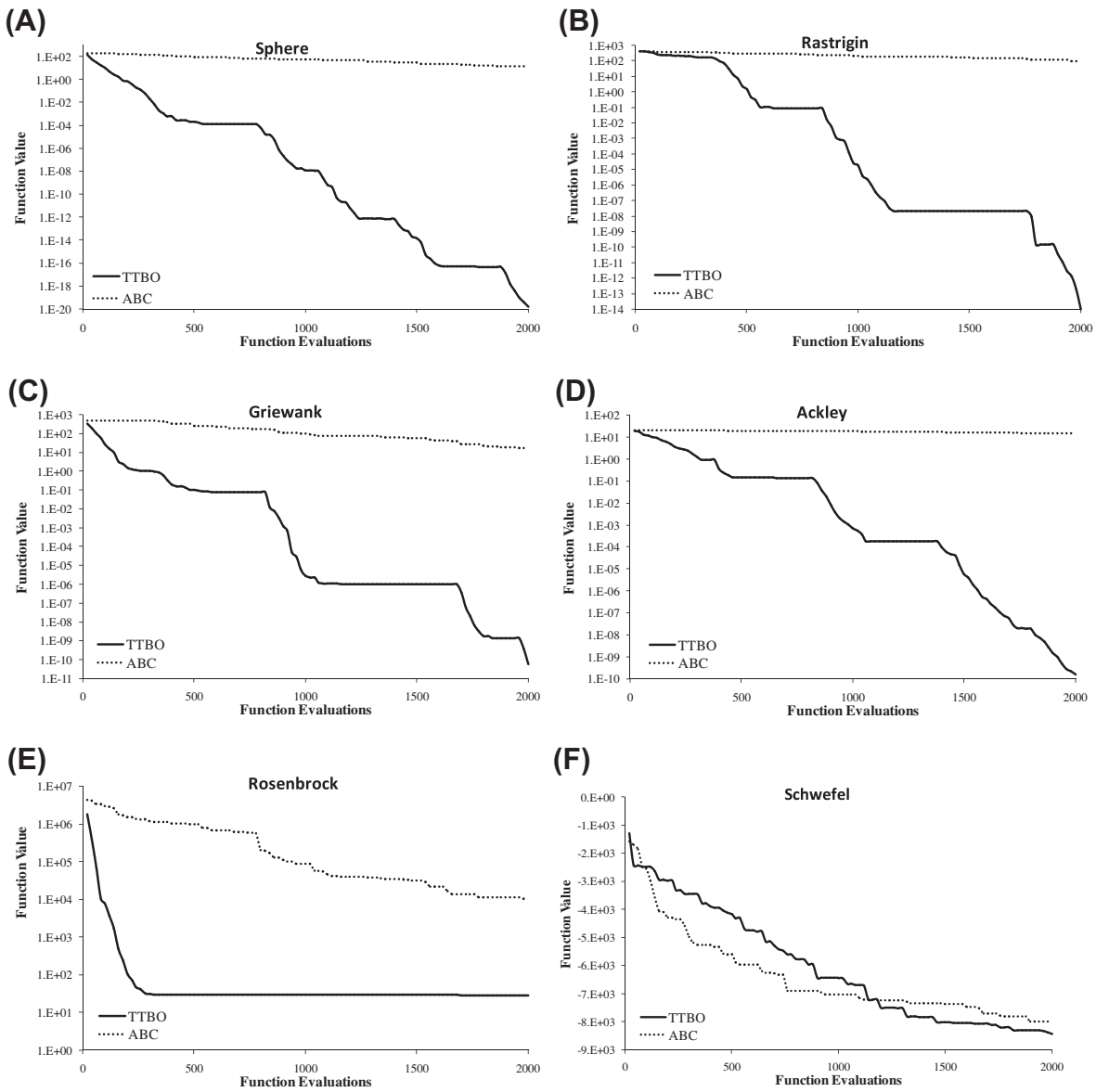

Fig. 6. Convergence curve for different benchmark problems considering continuous design variables: (A) Sphere, (B) Rastrigin, (C) Griewank, (D) Ackley, (E) Rosenbrock and (F) Schwefel.

seen from Table 5 that for function at s.no. 1, TLBO is better for mean number of function evaluations but the error value is better for NM-PSO. The TLBO method requires approximately 1/10th of function evaluations for the functions 1 and 2, 1/50th of function evaluations for the function 6, and 1/200th of function evaluations for the function 7. Also success rate of TLBO is better than NM- PSO for functions 4, 5, and 7. This experiment shows that TLBO is effective in terms of the computational effort, consistency, and obtaining the optimum solution.

5.3. Experiment 3

In this experiment, 5 different benchmark problems from Karaboga and Akay [14] are considered and comparison is made with the Harmony Search Algorithm (HS), Improved Bee Algorithm (IBA) and Artificial Bee Colony Optimization (ABC).

Comparison criteria are the mean solution and the standard solution for different independent runs. The mean solution describes the average ability of the algorithm to find the global solution and the standard deviation describes the variation in solution from the mean solution. In this experiment the algorithm runs for a specified maximum number of function eval-

uations. Moreover, the results are obtained for different independent runs and the values of the mean and standard deviation are calculated for the results obtained in different runs. Description of the functions is given in Table 6.

In this experiment, dimensions (D) of the benchmark functions are taken as 5, 10, 30, 50, and 100 for all the considered problems. Hence, experiment is performed for small scale to large scale problems. Maximum number of function evaluations was set as 50000 by Karaboga and Akay [14] for HS, IBA, and ABC. For TLBO, maximum number of function evaluations is set as 2000 (with population size of 10) for all the functions except Rosenbrock function for which its value is set as 50,000 (with population size of 50). Except Rosenbrock function, maximum number of function evaluations is 1/25th of the maximum number of function evaluations set for the HS, IBA and ABC algorithms. Table 7 shows the results of TLBO and the other considered algorithms. It can be seen from Table 7 that TLBO has outperformed all the algorithms except for Rosenbrock function. For Rosenbrock function, TLBO is still better than the HS and IBA algorithms. It is also seen from Table 7 that as the dimension increases to 100 for Rosenbrock function, TLBO gives better results than those given by the ABC algorithm. This experiment shows that TLBO is effective in finding the optimum solution with increase in dimensions.

5.4. Experiment 4

In this experiment, 8 different benchmark functions as given by Akay and Karaboga [2] are optimized and the results are compared with those given by PSO, DE, and ABC algorithms. Details are given in Table 8. A comparison criterion for this experiment is the mean of results obtained for different runs. This experiment is conducted for very high dimension of 500 for all the considered functions. In this experiment, the results are obtained for 30 independent runs. Maximum number of function evaluations considered by Akay and Karaboga [2] was 100,000. For TLBO, maximum number of function evaluations is set as 2000 (i.e. 1/50th of that given by Akay and Karaboga [2] with population size of 10) except for Rosenbrock, Schwefel and Penalised functions for which the maximum number of function evaluations is set as 100,000 (with population size of 10 for Schwefel and Penalised functions and 50 for Rosenbrock function). The results are given in Table 9. It can be seen from Table 9 that with only 2000 function evaluations, TLBO has outperformed all the algorithms for Sphere, Rastrigin and Ackley functions. For Step and Griewank functions, TLBO and ABC have shown the same result but TLBO requires only 1/50th of the function evaluations than those required by ABC. For the same number of function evaluations of 100,000, TLBO has shown better result for Rosenbrock function than all the algorithms. However, TLBO has shown inferior result for the Penalized function compared to that given by ABC, but still the result of TLBO is better than those given by PSO and DE algorithms. For Schwefel function, the results of TLBO and ABC are nearly the same. This experiment shows that TLBO is effective at very high dimensions for the functions having different characteristics like separability, multimodality, and regularity.

5.5. Experiment 5

This experiment is conducted to check the convergence rate of the TLBO algorithm. Comparison is made for TLBO and ABC. Six different benchmark functions are taken for the experiment viz. Sphere, Rastrigin, Rosenbrock, Griewank, Ackley and Schwefel with the dimension of 30. Maximum number of function evaluations is taken as 2000 with population size of 10 and maximum number of generations of 100. The results of ABC are obtained by using the code given in the website dedicated to ABC (http://mf.erciyes.edu.tr/abc/). A graph is plotted between the function value and the function evaluations. The function value considered is the average of function value for 10 different independent runs. Fig. 6 shows the convergence graphs for different benchmark problems. It is clear from Fig. 6 that the convergence rate of TLBO is higher than ABC for the considered problems except for Schwefel function for which the convergence is nearly the same.

6. Conclusions

All the nature-inspired algorithms such as GA, PSO, ACO, ABC, HS, etc. require algorithm parameters to be set for their proper working. Proper selection of parameters is essential for the searching of the optimum solution by these algorithms. A change in the algorithm parameters changes the effectiveness of the algorithms. To avoid this difficulty an optimization method, TLBO, which is algorithm parameter free, is presented in this paper. This method works on the effect of influence of a teacher on learners. Like other nature-inspired algorithms, TLBO is also a population based method which uses a population of solutions to proceed to the global solution. For TLBO, the population is considered as a group of learners or a class of learners. The process of working of TLBO is divided into two parts. The first part consists of 'Teacher Phase' and the second part consists of 'Learner Phase'. The 'Teacher Phase' means learning from the teacher and the 'Learner Phase' means learning through the interaction between learners.

The performance of the proposed TLBO method is checked with the recent and well-known optimization algorithms such as GA, ABC, PSO, HS, DE, Hybrid-PSO, etc. by experimenting with different benchmark problems with different characteristics like, multimodality, separability, regularity, and dimensionality. The effectiveness of TLBO method is also checked for different performance criteria, like success rate, mean solution, average function evaluations required, convergence rate, etc. The results show better performance of TLBO method over other natured-inspired optimization methods for the considered benchmark functions. Also, the TLBO method shows better performance with less computational efforts for the large scale

problems, i.e. problems with high dimensions. This method can be used for the optimization of engineering design applications.

Appendix A

1. De Jong function

$$\max F = 3905.93 - 100\left(x_1^2 - x_2\right)^2 - (1 - x_1)^2$$

2. Goldstein and Price

$$\min F = \left[1 + (x_1 + x_2 + 1)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]$$
$$\left[30 + (2x_1 - 3x_2)^2\left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$$

3. Martin and Gaddy

$$\min F = (x_1 - x_2)^2 + [(x_1 + x_2 - 10)/3]^2$$

4. Rosenbrock

$$\min F = \sum_{i=1}^{D}\left[100\left(x_i^2 - x_{i+1}\right)^2 + (1 - x_i)^2\right]$$

5. Sphere

$$\min F = \sum_{i=1}^{D} x_i^2$$

6. Powell badly scaled function

$$\max F = (10x_1x_2 - 1)^2 + [\exp(-x_1) + \exp(-x_2) - 1.0001]^2$$

7. B2 function

$$\min F = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$

8. Booth function

$$\min F = \left(x_1 + 2x_2 - 7\right)^2 + \left(2x_1 + x_2 - 5\right)^2$$

9. Griewank

$$f_{(x)} = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

10. Rastrigin

$$f_(x) = \sum_{i=1}^{D}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$$

11. Ackley

$$f_(x) = \sum_{i=1}^{D} -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{D}\cos 2\pi x_i\right)$$

12. Step

$$f_(x) = \sum_{i=1}^{D}[x_i + 0.5]^2$$

13. Schwefel

$$f_(x) = -\sum_{i=1}^{D}\left(x_i\sin\left(\sqrt{|x_i|}\right)\right),$$

14. Penalised

$$f(x) = \frac{\pi}{D}\left[10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i-1)^2\left\{1+10\sin^2(\pi y_{i+1})\right\} + (y_D-1)^2\right] + \sum_{i=1}^{D} u(x_i,10,100,4),$$

$$u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m, & x_i > a, \\ 0, & -a \leqslant x_i \leqslant a, \\ k(-x_i-a)^m, & x_i < -a \end{cases}$$

$$y_i = 1 + 1/4(x_i+1)$$

References

[1] A. Ahrari, A.A. Atai, Grenade explosion method – a novel tool for optimization of multimodal functions, Applied Soft Computing 10 (2010) 1132–1140.
[2] B. Akay, D. Karaboga, Artificial bee colony algorithm for large-scale problems and engineering design optimization, Journal of Intelligent Manufacturing (2010), doi:10.1007/s10845-010-0393-4.
[3] B. Akay, D. Karaboga, A modified artificial bee colony (ABC) algorithm for constrained optimization problems, Applied Soft Computing (2010), doi:10.1016/j.asoc.2010.12.001.
[4] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Information Sciences (2010), doi:10.1016/j.ins.2010.07.015.
[5] P. Chakraborty, S. Das, G.G. Roy, A. Abraham, On convergence of the multi-objective particle swarm optimizers, Information Sciences 181 (2011) 1411–1425.
[6] M. Dorigo, Optimization, Learning and Natural Algorithms, Ph.D. Thesis, Politecnico di Milano, Italy, 1992.
[7] M.M. Efrén, E.M.V. Mariana, D.C.G.R. Rubı, Differential evolution in constrained numerical optimization: an empirical study, Information Sciences 180 (2010) 4223–4262.
[8] M. Eusuff, E. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, Journal of Water Resources Planning and Management, ASCE 129 (2003) 210–225.
[9] J.D. Farmer, N. Packard, A. Perelson, The immune system, adaptation and machine learning, Physica D 22 (1986) 187–204.
[10] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2001) 60–70.
[11] Z.W. Geem, Novel derivative of harmony search algorithm for discrete design variables, Applied Mathematics and Computation 199 (2008) 223–230.
[12] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
[13] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, technical REPORT-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[14] D. Karaboga, B. Akay, Artificial bee colony (ABC), harmony search and bees algorithms on numerical optimization, Proceeding of IPROMS-2009 on Innovative Production Machines and Systems, Cardiff, UK, 2009.
[15] I. Karen, A.R. Yildiz, N. Kaya, N. Ozturk, F. Ozturk, Hybrid approach for genetic algorithm and Taguchi's method based design optimization in the automotive industry, International Journal of Production Research 4 (2006) 4897–4914.
[16] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942–1948.
[17] X. Li, J. Luo, M.R. Chen, N. Wang, An improved shuffled frog-leaping algorithm with external optimization for continuous optimization, Information Sciences (2010), doi:10.1016/j.ins.2010.07.016.
[18] J. Liu, L. Tang, A modified genetic algorithm for single machine scheduling, Computers & Industrial Engineering 37 (1999) 43–46.
[19] L. Liu, S. Yang, D. Wang, Force-imitated particle swarm optimization using the near-neighbor effect for locating multiple optima, Information Sciences 182 (2011) 139–155.
[20] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Systems Magazine 22 (2002) 52–67.
[21] R.V. Rao, Advanced Modeling and Optimization of Manufacturing Processes: International Research and Development, Springer Verlag, London, 2011.
[22] E. Rashedi, H.N. Pour, S. Saryazdi, GSA: a gravitational search algorithm, Information Sciences 179 (2009) 2232–2248.
[23] W. Shi, Q. Shen, W. Kong, B. Ye, QSAR analysis of tyrosine kinase inhibitor using modified ant colony optimization and multiple linear regression, European Journal of Medicinal Chemistry 42 (2007) 81–86.
[24] K.S.F. Shu, Z. Erwie, A hybrid simplex search and particle swarm optimization for unconstrained optimization, European Journal of Operational Research 181 (2007) 527–548.
[25] D. Simon, Biogeography-based optimization, IEEE Transactions on Evolutionary Computation 12 (2008) 702–713.
[26] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.
[27] C.L. Sun, J.C. Zeng, J.S. Pan, An improved vector particle swarm optimization for constrained optimization problems, Information Sciences 181 (2011) 1153–1163.
[28] Y. Wang, Y. Yang, Particle swarm optimization with preference order ranking for multi-objective optimization, Information Sciences 179 (2009) 1944–1959.
[29] A.R. Yildiz, A new design optimization framework based on immune algorithm and Taguchi method, Computers in Industry 60 (2009) 613–620.
[30] A.R. Yildiz, A novel hybrid immune algorithm for global optimization in design and manufacturing, Robotics and Computer-Integrated Manufacturing 25 (2009) 261–270.
[31] A.R. Yildiz, A novel particle swarm optimization approach for product design and manufacturing, International Journal of Advanced Manufacturing Technology 40 (2009) 617–628.
[32] A.R. Yıldız, N. Ozturk, N. Kaya, F. Ozturk, Hybrid multi-objective shape design optimization using Taguchi's method and genetic algorithm, Structural and Multidisciplinary Optimization 34 (2007) 277–365.
[33] A.R. Yıldız, N. Ozturk, N. Kaya, F. Ozturk, Integrated optimal topology design and shape optimization using neural networks, Structural and Multidisciplinary Optimization 25 (2003) 251–260.