Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: http://www.tandfonline.com/loi/geno20

Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems

R. V. Rao , V. J. Savsani & J. Balic

To cite this article: R. V. Rao , V. J. Savsani & J. Balic (2012) Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems, Engineering Optimization, 44:12, 1447-1462, DOI: 10.1080/0305215X.2011.652103

To link to this article: https://doi.org/10.1080/0305215X.2011.652103

Published online: 29 Mar 2012.

Engineering Optimization
Vol. 44, No. 12, December 2012, 1447–1462

Taylor & Francis
Taylor & Francis Group

Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems

R.V. Rao[a]*, V. J. Savsani[a] and J. Balic[b]

[a]*Department of Mechanical Engineering, S.V. National Institute of Technology, Surat, Gujarat, India;* [b]*Faculty of Mechanical Engineering, University of Maribor, Slovenia*

(*Received 31 August 2011; final version received 24 November 2011*)

An efficient optimization algorithm called teaching–learning-based optimization (TLBO) is proposed in this article to solve continuous unconstrained and constrained optimization problems. 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 algorithm is tested on 25 different unconstrained benchmark functions and 35 constrained benchmark functions with different characteristics. For the constrained benchmark functions, TLBO is tested with different constraint handling techniques such as superiority of feasible solutions, self-adaptive penalty, ε-constraint, stochastic ranking and ensemble of constraints. The performance of the TLBO algorithm is compared with that of other optimization algorithms and the results show the better performance of the proposed algorithm.

Keywords: real-parameter optimization; unconstrained benchmark functions; constrained benchmark functions; teaching–learning-based optimization

1. Introduction

Constrained and unconstrained optimization problems are generally associated with many difficulties such as multi-modality, dimensionality and differentiability. Traditional optimization techniques generally fail to solve such problems, especially with nonlinear objective functions. To overcome these difficulties, there is a need to develop more powerful optimization techniques and research is continuing to find effective optimization techniques. Some of the well-known population-based optimization techniques developed during the past three decades are: genetic algorithm (GA) (Holland 1975), artificial immune algorithm (AIA) (Farmer *et al.* 1986), ant colony optimization (ACO) (Dorigo 1992), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), differential evolution (DE) (Storn and Price 1997), harmony search (HS) (Geem *et al.* 2001), bacteria foraging optimization (BFO) (Passino 2002), shuffled frog leaping (SFL) (Euusuff and Lansey 2003), artificial bee colony (ABC) (Karaboga 2005), biogeography-based optimization (BBO) (Simon 2008), gravitational search algorithm (GSA) (Rashedi *et al.* 2009) and grenade

*Corresponding author. Email: ravipudirao@gmail.com

ISSN 0305-215X print/ISSN 1029-0273 online

http://dx.doi.org/10.1080/0305215X.2011.652103
http://www.tandfonline.com

explosion method (GEM) (Ahrari and Atai 2010). These algorithms have been applied to many engineering optimization problems and have proved effective in solving specific kinds of problem.

Proper selection of the parameters is essential for the searching of the optimum solution by the above-mentioned optimization algorithms. A change in the algorithm parameters influences the effectiveness of the algorithm. In addition to the population size as a controlling parameter, the existing optimization algorithms have their specific controlling parameters. The most commonly used evolutionary optimization technique is GA. However, GA provides a near optimal solution for a complex problem with a large number of variables and constraints. This is mainly due to the difficulty in determining the optimum controlling parameters such as crossover rate and mutation rate. The same is the case with PSO, which uses inertia weight, and social and cognitive parameters. Similarly, ABC requires optimum controlling parameters of the number of bees (employed, scout and onlookers) and limit. HS requires harmony memory consideration rate, pitch adjusting rate and the number of improvizations. SFLA requires the number of memeplexes and iteration per memeplex. ACO requires exponent parameters, pheromone evaporation rate and reward factor. Sometimes, the difficulty in the selection of parameters increases with modifications and hybridization. Therefore, efforts must be continued to develop an optimization technique that is free from the algorithm-specific 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 article to obtain global solutions for continuous nonlinear functions with less computational effort. The TLBO method works on the philosophy of teaching and learning.

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

2. Teaching–learning-based optimization algorithm

TLBO is a teaching–learning process-inspired algorithm proposed by Rao *et al.* (2011, 2012) based on the influence of a teacher on the output of learners in a class. The algorithm mimics the teaching–learning ability of the teacher and learners in a classroom. The teacher and learners are the two vital components of the algorithm. The algorithm describes two basic modes of the learning: (1) through teacher (known as the teacher phase) and (2) interacting with the other learners (known as the learner phase). The output in the TLBO algorithm is considered in terms of the results or grades of the learners, which depend on the quality of the teacher. A high-quality teacher is usually considered a highly learned person who trains learners so that they can achieve better results in terms of their marks or grades. Moreover, learners also learn through interaction among themselves, which also helps to improve their results.

TLBO is a population-based method. In this optimization algorithm, a group of learners, n, is considered as a population and different subjects offered to the learners are considered as different design variables, m, of the optimization problem. A learner's result is analogous to the 'fitness' value of the optimization problem. The best solution in the entire population is considered as the teacher. The design variables are actually the parameters involved in the objective function of the given optimization problem and the best solution is the best value of the objective function. TLBO is divided into two parts: the teacher phase and the learner phase. The working of both of these phases is explained below.

2.1. *Teacher phase*

This is the first part of the algorithm, where learners learn through the teacher. During this phase a teacher tries to increase the mean result of the class in the subject taught by him or her, depending on his or her capability.

The teacher is generally considered the most learned person in the society. When a teacher teaches in a classroom it is expected that he or she knows more than the students. This aspect is considered for the TLBO algorithm. This algorithm is a population-based method which starts with a set of solutions known as the population. As the teacher is considered the most learned person, the best solution (defined by its objective function value) from the population is considered as the teacher. The purpose of a teacher is to increase the knowledge of the students. In practice, however, it is not possible for a teacher to increase the level of the students equally as different students possess different knowledge levels. A survey of the knowledge possessed by the students will give some mean knowledge value. A teacher, after teaching, will improve this mean level of the students to a better mean knowledge value. This aspect is represented in mathematical form and is implemented for the optimization.

Any value in the solution is represented as $X_{j,k,i}$ in this article, where j means the jth design variable (*i.e.* subject taken by the learners), $j = 1, 2, \ldots, m$; k represents the kth population member (*i.e.* learner), $k = 1, 2, \ldots, n$; and i represents the ith iteration, $i = 1, 2, \ldots, G_{max}$, where G_{max} is the number of maximum generations (iterations).

The teacher phase starts by identifying the teacher (*i.e.* best solution) from the population. The best solution is determined from the objective function value. If the problem is one of minimization then the solution that gives the minimum value of objective function will be considered as the best solution. Let $X_{k,i}$ be that best solution at any iteration i for which the value of $f(X_{k,i})$ is minimum among the population. This best solution will be represented as $X_{kbest,i}$. The next step is to calculate the mean result $M_{j,i}$ of the learners in a particular subject j. As mentioned earlier, a teacher tries to increase the mean result of the class in the subject taught by him or her depending on his or her capability. The increase in the existing mean result of each subject by the teacher for each subject is given by:

$$Difference\_Mean_{j,k,i} = r_{j,i}(X_{j,kbest,i} - T_F M_{j,i}) \tag{1}$$

where $X_{j,kbest,i}$ is the result of the best learner (*i.e.* teacher) in subject j, T_F is the teaching factor which decides the value of mean to be changed (capability of a teacher), and $r_{j,i}$ is the random number in the range [0,1]. The value of T_F can be either 1 or 2. The value of T_F is decided randomly with equal probability as:

$$T_F = round[1 + rand(0,1)\{2\text{-}1\}] \tag{2}$$

T_F is not a parameter of the TLBO algorithm. The value of T_F is not given as an input to the algorithm and its value is randomly decided by the algorithm using Equation (2). Twelve different benchmark functions from CEC 2006 (Liang *et al.* 2006) are demonstrated using different values of T_F. For the demonstration, values of T_F are taken as 0, 1, 2, 3, 4, 5 and that given by Equation (2). Different values of T_F are experimented with for 25 runs and the mean solution is recorded for the comparison. It is observed that the algorithm performs better for the values of $T_F = 1$ and 2, but it performs much better if the value of T_F varies as per Equation (2), *i.e.* randomly, either value 1 or 2. Hence, to simplify the algorithm, it is suggested that the teaching factor takes a value of either 1 or 2 depending on the rounding-up criteria given by Equation (2).

Based on the $Difference\_Mean_{j,k,i}$, the existing solution is updated in the teacher phase according to the following expression:

$$X'_{j,k,i} = X_{j,k,i} + Difference\_Mean_{j,k,i} \tag{3}$$

where $X'_{j,k,i}$ is the updated value of $X_{j,k,i}$. Accept $X'_{j,k,i}$ if it gives w better function value than $X_{j,k,i}$. All the accepted function values at the end of the teacher phase are maintained and these values become the input to the learner phase.

2.2. *Learner phase*

This is the second part of the algorithm, where learners increase their knowledge by interaction among themselves. A learner interacts randomly with other learners to enhance his or her knowledge. A learner learns new things if the other learners have more knowledge than he or she does. Considering a population size of n, the learning phenomenon of this phase is expressed below.

At any iteration i, each learner is compared with the other learners randomly. For comparison, randomly select two learners P and Q such that $X'_{P,i} \neq X'_{Q,i}$ (where $X'_{P,i}$ and $X'_{Q,i}$ are the updated values at the end of the teacher phase):

$$X''_{j,P,i} = X'_{j,P,i} + r_{j,i}(X'_{j,P,i} - X'_{j,Q,i}), \text{ if } f(X'_{P,i}) < f(X'_{Q,i}) \tag{4a}$$

$$X''_{j,P,i} = X'_{j,P,i} + r_{j,i}(X'_{j,Q,i} - X'_{j,P,i}), \text{ if } f(X'_{Q,i}) < f(X'_{P,i}) \tag{4b}$$

Accept $X''_{j,P,i}$ if it gives a better function value.

The learner phase is implemented using the following loops at any iteration i:

For k =1:n

Let the present learner be $X'_{P,i}$.

Randomly select another learner $X'_{Q,i}$, such that $X'_{P,i} \neq X'_{Q,i}$

If $f(X'_{P,i}) < f(X'_{Q,i})$,

For j=1:m; $X''_{j,P,i} = X'_{j,P,i} + r_{j,i}(X'_{j,P,i} - X'_{j,Q,i})$; *End For*

Else; For j=1:m; $X''_{j,P,i} = X'_{j,P,i} + r_{j,i}(X'_{j,Q,i} - X'_{j,P,i})$; *End For*;

End If

End For

All the accepted function values at the end of the learner phase are maintained and these values become the input to the teacher phase of the next iteration. The values of $r_{j,i}$ used in Equations (1) and (4) can be different.

The flowchart for the TLBO algorithm is shown in Figure 1.

3. Implementation of TLBO for the real-parameter optimization

The TLBO algorithm described by Rao *et al.* (2011, 2012) is extended here for the complex unconstrained and constrained real-parameter optimization problems. For the constrained benchmark functions, TLBO is tested with different constraint handling techniques such as superiority of feasible solutions, self-adaptive penalty, ε-constraint, stochastic ranking and ensemble of constraint handling.

3.1. *Experiment 1*

It is common practice in the field of optimization to compare different algorithms using different benchmark problems. These comparisons are limited to the test problems taken for the study and sometimes the chosen algorithm and the test problems are complementary to each other and the same algorithm may not show the same performance for the other real-parameter optimization problems. So, a common platform is required to compare the performance of different algorithms for different benchmark problems. Das *et al.* (2011) presented a comprehensive review of the basic concepts related to real-parameter evolutionary multimodal optimization, a survey of the major evolutionary optimization techniques, a detailed account of the adaptation of evolutionary algorithms from diverse paradigms to tackle multimodal problems, benchmark problems and performance measures. Zhou *et al.* (2011) surveyed the development of multi-objective evolutionary algorithms and presented various algorithmic frameworks.

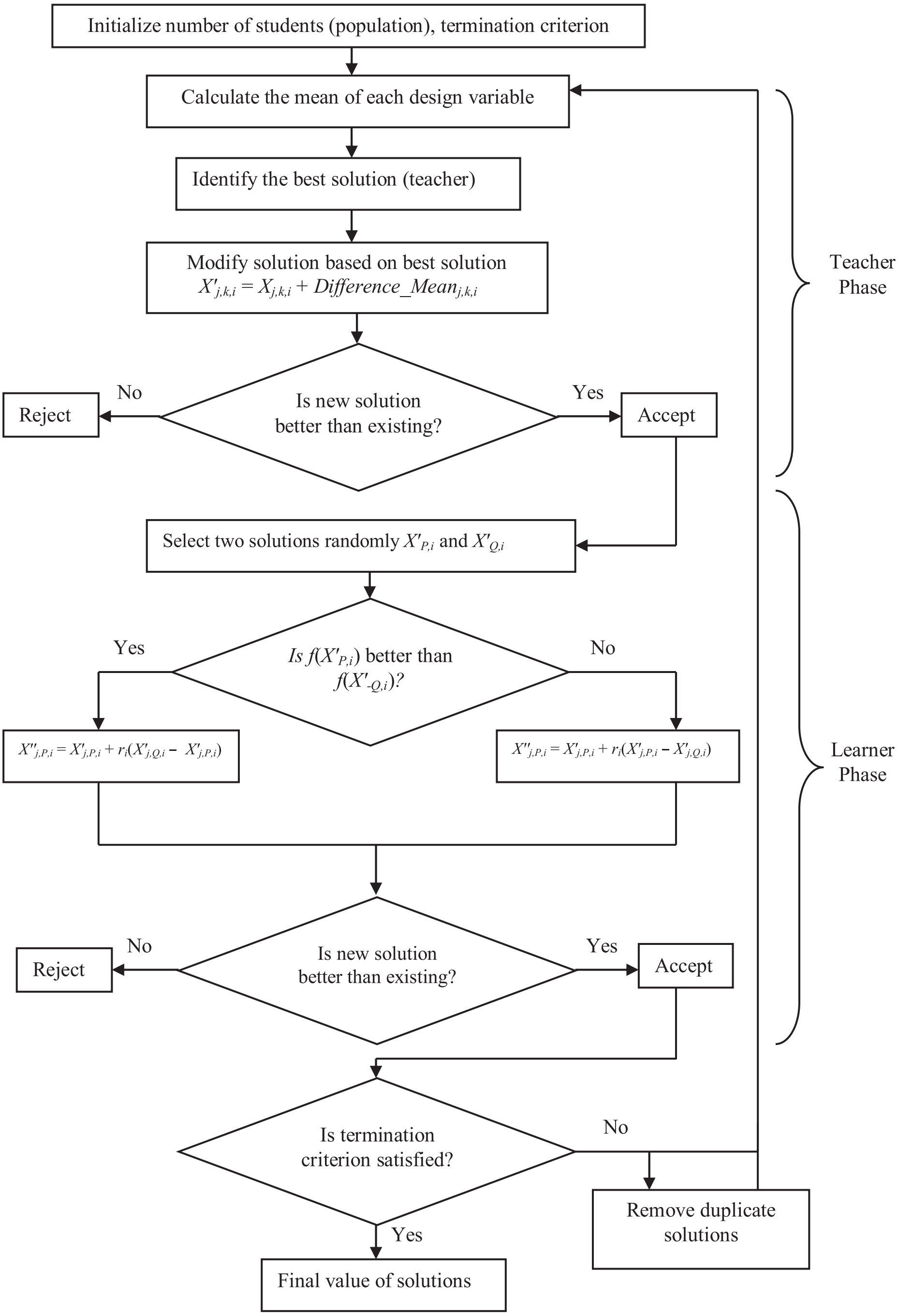

Figure 1. Flowchart of the teaching–learning-based optimization (TLBO) algorithm.

The Congress on Evolutionary Computation 2005 (CEC 2005) (Suganthan *et al.* 2005) provided a common platform for comparison of the performance of different algorithms by specifying a common termination criterion, size of problem, initialization scheme, *etc*. In this article, 25 different benchmark problems are experimented with using the proposed TLBO algorithm. These benchmark functions have different characteristics such as separability, scalability and multimodality. 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. 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. Furthermore, complexity increases with the increase in dimensionality. All the benchmark functions proposed in CEC 2005 are generated from the basic benchmark functions (Sphere, Rastrigin, Rosenbrock, Schwefel, Griewank, *etc.*) by shifting, rotating or hybridizing different basic benchmark functions. Shifting, rotating and hybridizing add more complexity to the benchmark functions and so testing of algorithms for such problems is a real challenge. These benchmark functions are available at http://www.ntu.edu.sg/home/EPNSugan.

Some common evolution criteria were presented in CEC 2005 (Suganthan *et al.* 2005) and these criteria are considered in this article for completeness. All the benchmark functions are run for 25 times each. The function error value $(f(x) - f(x^*))$ is recorded after 1E3, 1E4, 1E5 number of function evaluations and at termination for each run. Function error is considered as the difference between the global optimum and the best result obtained by the algorithm. If the algorithm finds the global optimum, then the error will be 0E0. Termination is done when the maximum function evaluations equals to $10,000^*D$, where D indicates the dimension of the problem. Dimensions taken for the experiment are 10 and 30. Function error values are recorded for all 25 runs and sorted from the best value to the worst value. After sorting, 1st (best), 7th, 13th (median), 19th and 25th (worst) values are recorded. Mean and standard deviation for all the runs are also recorded. TLBO is coded in MATLAB 7 and run on a laptop computer with a 2.2 GHz Intel Pentium processor with 3 GB RAM. TLBO is applied for all the benchmark functions by considering the population size of 20. As TLBO is a parameter-less algorithm, no other parameter is required for the working of the algorithm. Function error values for dimension 30 are presented in Tables 1–3. The procedure to calculate the complexity of the algorithm is given in Suganthan *et al.* (2005). The complexity of TLBO is calculated as $(T2_{(\mathrm{mean})} - T1)/T0$, where $T0$ is the time to calculate the following:

For i=1:1000000

Table 1. Error values achieved in 1.00E+03, 1.00E+04, 1.00E+05 and 3.00E+05 function evaluations for functions B01–B08 with dimension $D = 30$.

$D = 30$		B01	B02	B03	B04	B05	B06	B07	B08
1.00E+03	Best	5.43E+03	6.39E+04	3.47E+06	3.96E+04	1.24E+04	5.23E+08	4.83E+00	2.09E+01
	Median	1.23E+04	8.21E+04	7.93E+07	4.79E+04	1.84E+04	1.46E+09	1.57E+01	2.12E+01
	Worst	1.97E+04	1.10E+05	2.33E+08	5.98E+04	2.65E+04	3.33E+09	2.35E+01	2.13E+01
	Mean	1.29E+04	8.43E+04	8.02E+07	4.77E+04	1.85E+04	1.45E+09	1.46E+01	2.11E+01
	Std	5.75E+03	1.76E+04	7.78E+07	7.79E+03	4.67E+03	1.11E+09	6.65E+00	1.62E−01
1.00E+04	Best	3.67E+00	1.23E+02	4.48E+05	3.24E+04	3.52E+03	9.83E+03	4.45E−01	2.09E+01
	Median	5.29E+00	1.87E+02	9.22E+05	9.32E+04	9.22E+03	1.89E+04	9.32E−01	0.10E+01
	Worst	9.33E+00	2.56E+02	2.87E+06	1.32E+05	1.82E+04	2.45E+04	3.43E+00	2.13E+01
	Mean	5.28E+00	1.92E+02	9.12E+05	9.00E+04	9.87E+04	1.76E+04	1.35E+00	2.10E+01
	Std	2.77E+00	3.20E+01	9.43E+05	3.81E+04	5.50E+03	5.94E+03	1.20E+00	1.64E−01
1.00E+05	Best	0.00E+00	0.00E+00	1.76E+05	3.34E+02	2.24E+03	4.46E+01	3.40E−03	2.08E+01
	Median	0.00E+00	7.34E−02	1.94E+05	1.65E+03	8.96E+03	1.23E+02	8.72E−03	2.09E+01
	Worst	0.00E+00	1.84E−01	2.87E+05	9.21E+03	1.93E+04	2.13E+02	3.42E−02	2.10E+01
	Mean	0.00E+00	7.59E−02	1.96E+05	1.67E+03	9.26E+03	1.24E+02	8.70E−03	2.09E+01
	Std	0.00E+00	6.30E−02	4.06E+04	3.47E+03	5.66E+03	6.28E+01	1.14E−02	6.32E−02
3.00E+05	Best	0.00E+00	0.00E+00	1.76E+05	0.00E+00	2.17E+03	1.18E+01	1.28E−06	2.08E+01
	Median	0.00E+00	0.00E+00	1.81E+05	0.00E+00	4.28E+03	1.84E+01	3.44E−03	2.08E+01
	Worst	0.00E+00	0.00E+00	1.88E+05	0.00E+00	7.95E+03	6.93E+01	4.49E−02	2.09E+01
	Mean	0.00E+00	0.00E+00	1.82E+05	0.00E+00	4.60E+03	3.47E+01	1.63E−02	2.08E+01
	Std	0.00E+00	0.00E+00	4.58E+03	0.00E+00	1.98E+03	2.42E+01	1.89E−02	4.90E−02

Note: Std = standard deviation.

Table 2. Error values achieved in 1.00E+03, 1.00E+04, 1.00E+05 and 3.00E+05 function evaluations for functions B09–B17 with dimension $D = 30$.

$D = 30$		B09	B10	B11	B12	B13	B14	B15	B16	B17
1.00E+03	Best	1.65E+02	3.42E+02	4.36E+01	6.34E+05	4.47E+01	1.39E+01	6.13E+02	4.58E+02	4.38E+02
	Median	2.13E+02	4.17E+02	4.69E+02	7.75E+05	5.18E+01	1.41E+01	7.93E+02	4.96E+02	5.26E+02
	Worst	2.67E+02	4.44E+02	5.01E+02	8.99E+05	1.34E+02	1.42E+01	8.72E+02	6.24E+02	5.92E+02
	Mean	2.13E+02	4.03E+02	3.90E+02	7.73E+05	7.07E+01	1.41E+01	7.72E+02	5.17E+02	5.08E+02
	Std	3.59E+01	3.70E+01	1.74E+02	9.33E+04	3.32E+01	1.10E−01	8.84E+01	5.76E+01	5.76E+01
1.00E+04	Best	9.80E+01	1.96E+02	3.42E+02	2.34E+05	1.15E+01	1.35E+01	3.93E+02	3.12E+02	2.77E+02
	Median	1.19E+02	2.26E+02	3.82E+02	3.01E+05	1.48E+01	1.39E+01	5.13E+02	4.13E+02	3.34E+02
	Worst	1.86E+02	2.41E+02	4.12E+02	3.36E+05	1.64E+01	1.40E+01	6.32E+02	4.97E+02	3.52E+02
	Mean	1.27E+02	2.24E+02	3.81E+02	2.96E+05	1.42E+01	1.38E+01	5.17E+02	4.17E+02	3.27E+02
	Std	3.47E+01	1.83E+01	2.83E+01	4.04E+04	2.12E+00	1.92E−01	9.62E+01	7.31E+01	3.01E+01
1.00E+05	Best	5.69E+01	1.48E+02	3.42E+01	7.24E+04	8.21E+00	1.31E+01	2.00E+02	2.63E+02	2.64E+02
	Median	8.63E+01	1.97E+02	3.53E+01	1.32E+05	9.87E+00	1.34E+01	3.00E+02	3.18E+02	0.84E+02
	Worst	9.01E+01	2.36E+02	3.99E+01	1.58E+05	1.14E+01	1.36E+01	4.00E+02	3.41E+02	3.33E+02
	Mean	7.79E+01	1.93E+02	3.63E+01	1.22E+05	1.00E+01	1.34E+01	3.00E+02	3.08E+02	2.89E+02
	Std	1.34E+01	3.19E+01	2.14E+00	3.03E+04	1.20E+00	1.60E−01	8.94E+01	2.81E+01	2.60E+01
3.00E+05	Best	2.12E+01	6.16E+01	1.38E+01	1.43E+03	1.93E+00	1.22E+01	2.00E+02	2.52E+02	2.64E+02
	Median	2.31E+01	9.96E+01	1.82E+01	5.64E+03	2.67E+00	1.32E+01	3.00E+02	2.83E+02	2.64E+02
	Worst	2.46E+01	1.78E+02	2.16E+01	5.89E+04	4.83E+00	1.34E+01	4.00E+02	3.24E+02	2.95E+02
	Mean	2.30E+01	1.09E+02	1.77E+01	1.84E+04	3.01E+00	1.30E+01	2.80E+02	2.31E+02	2.73E+02
	Std	1.14E+00	4.02E+01	2.71E+00	2.17E+04	1.02E+00	4.27E−01	7.48E+01	1.17E+02	1.21E+01

Note: Std = standard deviation.

Table 3. Error values achieved in 1.00E+03, 1.00E+04, 1.00E+05 and 3.00E+05 function evaluations for functions B18–B25 with dimension $D = 30$.

$D = 30$		B18	B19	B20	B21	B22	B23	B24	B25
1.00E+03	Best	1.20E+03	1.17E+03	1.20E+03	1.14E+03	1.43E+03	1.23E+03	2.03E+02	2.22E+02
	Median	1.32E+03	1.23E+03	1.31E+03	1.23E+03	1.62E+03	1.26E+03	2.04E+02	2.73E+02
	Worst	1.42E+03	1.40E+03	1.38E+03	1.28E+03	1.88E+03	1.29E+03	2.06E+02	2.89E+02
	Mean	1.32E+03	1.25E+03	1.30E+03	1.22E+03	1.65E+03	1.26E+03	2.04E+02	2.62E+02
	Std	7.23E+01	7.86E+01	6.31E+01	4.77E+01	1.52E+02	1.94E+01	1.17E+00	2.55E+01
1.00E+04	Best	9.03E+02	9.89E+02	9.89E+02	5.00E+02	9.06E+02	7.28E+02	2.01E+02	2.00E+02
	Median	1.01E+03	1.03E+03	1.04E+03	6.30E+02	1.10E+03	8.14E+02	2.01E+02	0.03E+02
	Worst	1.03E+03	1.05E+03	1.07E+03	8.60E+02	1.18E+03	8.87E+02	2.01E+02	2.05E+02
	Mean	9.97E+02	1.02E+03	1.03E+03	6.54E+02	1.07E+03	8.07E+02	2.01E+02	2.02E+02
	Std	5.33E+01	2.63E+01	3.39E+01	1.41E+02	1.10E+02	7.06E+01	0.00E+00	2.17E+00
1.00E+05	Best	9.06E+02	9.06E+02	9.06E+02	5.00E+02	8.91E+02	5.34E+02	2.01E+02	2.00E+02
	Median	9.07E+02	9.63E+02	9.08E+02	5.00E+02	9.19E+02	5.56E+02	2.01E+02	2.00E+02
	Worst	9.70E+02	1.05E+03	1.04E+02	6.44E+02	9.31E+02	5.97E+02	2.01E+02	2.00E+02
	Mean	9.19E+02	9.77E+02	7.56E+02	5.41E+02	9.16E+02	5.59E+02	2.01E+02	2.00E+02
	Std	2.54E+01	4.90E+01	3.27E+02	5.66E+01	1.41E+01	2.17E+01	3.89E−14	0.00E+00
3.00E+05	Best	9.08E+02	9.08E+02	9.03E+02	5.00E+02	8.60E+02	5.34E+02	2.00E+02	2.00E+02
	Median	9.08E+02	9.08E+02	9.08E+02	5.00E+02	9.01E+02	5.36E+02	2.01E+02	2.00E+02
	Worst	9.09E+02	9.10E+02	9.09E+02	5.05E+02	9.04E+02	5.56E+02	2.01E+02	2.00E+02
	Mean	9.08E+02	9.09E+02	9.07E+02	5.01E+02	8.89E+02	5.39E+02	2.01E+02	2.00E+02
	Std	4.90E−01	8.00E−01	2.24E+00	1.96E+00	1.76E+01	8.45E+00	4.00E−01	0.00E+00

Note: Std = standard deviation.

x=(double) 5.55
*x=x+x;x=x/2;x=x*x;x=sqrt(x);x=ln(x);x=exp(x);y=x/x;*
end

$T1$ is the time to calculate only function B03 for 200,000 evaluations for a certain dimension and T2 is the mean time for the optimization algorithm to calculate function B03 for 200,000 function evaluations for the same dimension. $T2_{\text{(mean)}}$ is the mean time for T2 obtained for five times. The complexity of the algorithm is 3581.2, 3628.8 and 3834.0 for $D = 10$, $D = 30$ and $D = 50$, respectively.

The performance of TLBO is compared with seven other optimization algorithms: PSO-recombination with dynamic linkage (PSO-RDL) (Jian 2006), PSO-dynamic multiswarm optimization (DMS-PSO) (Liang and Suganthan 2005), SPC-PNX (Ballester *et al.* 2005), DE (Ronkkonen *et al.* 2005), self-adaptive DE (Quin and Suganthan 2005), restart CMA-ES (Auger and Hansen 2005) and ABC (Akay and Karaboga 2010). The algorithms are compared based on the mean value for 1E5 function evaluations with dimension 10. It can be seen from Tables 4–6 that

Table 4. Comparison of teaching–learning-based optimization (TLBO) with other state-of-the-art optimization algorithms for functions B01–B08 with dimension $D = 10$ and 1.00E+05 function evaluations.

Algorithm	B01	B02	B03	B04	B05	B06	B07	B08
PSO-RDL[a]	2.50E−14	1.77E−13	9.68E−02	2.47E−07	2.09E−07	9.57E−01	5.73E−02	2.00E+01
DMS-PSO[b]	0.00E+00	1.30E−13	7.01E−09	1.89E−03	1.14E−06	6.89E−08	4.52E−02	2.00E+01
SPC-PNX[c]	8.90E−09	9.63E−09	1.08E+05	9.38E−09	9.15E−09	1.89E+01	8.26E−02	2.10E+01
DE[d]	0.00E+00	0.00E+00	1.94E−06	9.09E−15	0.00E+00	1.59E+01	1.46E−01	2.04E+01
SaDE[e]	0.00E+00	1.05E−13	1.67E−05	1.42E−05	1.23E−02	1.20E−08	1.99E−02	2.00E+01
RESTART CMA-ES[f]	5.20E−09	4.70E−09	5.60E−09	5.02E−09	6.58E−09	4.87E−09	3.31E−09	2.00E+01
ABC[g]	4.89E−17	4.81E−17	2.50E+03	1.50E−16	5.82E+01	3.31E+00	2.52E−01	2.03E+01
TLBO[h]	0.00E+00	0.00E+00	2.86E+03	0.00E+00	0.00E+00	4.10E−02	1.26E−02	2.00E+01

Note: Algorithm: [a]Jian (2006); [b]Liang and Suganthan (2005); [c]Ballester et al. (2005); [d]Ronkkonen et al. (2005); [e]Qin and Suganthan (2005); [f]Auger and Hansen (2005); [g]Akay and Karaboga (2010); [h]TLBO.

Table 5. Comparison of teaching–learning-based optimization (TLBO) with other state-of-the-art optimization algorithms for functions B09–B17 with dimension $D = 10$ and 1.00E+05 function evaluations.

Algorithm	B09	B10	B11	B12	B13	B14	B15	B16	B17
PSO-RDL[a]	1.25E+01	3.86E+01	5.58E+00	1.31E+02	8.87E−01	3.78E+00	2.71E+02	2.20E+02	2.22E+02
DMS-PSO[b]	0.00E+00	3.62E+00	4.62E+00	2.40E+00	3.69E−01	2.36E+00	4.85E+00	9.48E+01	1.10E+02
SPC-PNX[c]	4.02E+00	7.30E+00	1.91E+00	2.60E+02	8.38E−01	3.05E+00	2.54E+02	1.10E+02	1.19E+02
DE[d]	9.55E−01	1.25E+01	8.47E−01	3.17E+01	9.77E−01	3.45E+00	2.59E+02	1.13E+02	1.15E+02
SaDE[e]	0.00E+00	4.97E+00	4.89E+00	4.50E−07	2.20E−01	2.92E+00	3.20E+01	1.01E+02	1.14E+02
RESTART CMA-ES[f]	2.39E−01	7.96E−02	9.34E−01	2.93E+01	6.96E−01	3.01E+00	2.28E+02	9.13E+01	1.23E+02
ABC[g]	4.87E−17	2.22E+01	5.46E+00	9.85E+01	2.96E−02	3.41E+00	1.53E−01	1.75E+02	1.96E+02
TLBO[h]	9.92E−01	1.17E+01	3.88E+00	2.35E+03	2.14E−01	2.67E+00	8.60E+01	9.06E+01	1.07E+02

Note: Algorithm: [a]Jian (2006); [b]Liang and Suganthan (2005); [c]Ballester et al. (2005); [d]Ronkkonen et al. (2005); [e]Qin and Suganthan (2005); [f]Auger and Hansen (2005); [g]Akay and Karaboga (2010); [h]TLBO.

Table 6. Comparison of teaching–learning-based optimization (TLBO) with other state-of-the-art optimization algorithms for functions B18–B25 with dimension $D = 10$ and 1.00E+05 function evaluations.

Algorithm	B18	B19	B20	B21	B22	B23	B24	B25
PSO-RDL[a]	1.02E+03	9.85E+02	9.59E+02	9.94E+02	8.87E+02	1.08E+03	7.20E+02	1.76E+03
DMS-PSO[b]	7.61E+02	7.14E+02	8.22E+02	5.36E+02	6.92E+02	7.30E+02	2.24E+02	3.66E+02
SPC-PNX[c]	4.40E+02	3.80E+02	4.40E+02	6.80E+02	7.49E+02	5.76E+02	2.00E+02	4.06E+02
DE[d]	4.00E+02	4.20E+02	4.60E+02	4.92E+02	7.18E+02	5.72E+02	2.00E+02	9.23E+02
SaDE[e]	7.19E+02	7.05E+02	7.13E+02	4.64E+02	7.32E+02	6.64E+02	2.00E+02	3.76E+02
RESTART CMA-ES[f]	3.32E+02	2.26E+02	3.00E+02	5.00E+02	7.29E+02	5.59E+02	2.00E+02	3.74E+02
ABC[g]	4.46E+02	4.51E+02	4.38E+02	4.07E+02	8.59E+02	4.98E+02	2.02E+02	2.00E+02
TLBO[h]	5.02E+02	3.77E+02	4.26E+02	4.06E+02	5.29E+02	5.95E+02	2.00E+02	5.77E+02

Note: Algorithm: [a]Jian (2006); [b]Liang and Suganthan (2005); [c]Ballester et al. (2005); [d]Ronkkonen et al. (2005); [e]Qin and Suganthan (2005); [f]Auger and Hansen (2005); [g]Akay and Karaboga (2010); [h]TLBO.

TLBO has outperformed all the other algorithms or performed equally best for nine benchmark functions: B01, B02, B04, B05, B16, B17, B21, B22 and B24. Moreover, TLBO has outperformed six other algorithms for six other benchmark functions: B07, B08, B13, B14, B19 and B20.

3.2. *Experiment 2*

In this experiment, 22 different constrained benchmark functions from CEC 2006 (Liang *et al.* 2006) are experimented with. The capability of the algorithm to find the global solution for the constrained problem depends on the constraint handling technique. In this experiment, four different constraint handling techniques: superiority of feasible solutions (SF) (Deb 2000), self-adaptive penalty approach (SP) (Tessema and Yen 2006), ε-constraint technique (EC) (Takahama and Sakai 2006) and stochastic ranking technique (SR) (Runarsson and Yao 2000), are experimented with using TLBO. Moreover, a new constraint handling technique, ensemble of constraint handling technique (ECHT), suggested in Mallipeddi and Suganthan (2010), is also experimented with using TLBO. The technique suggested by Mezura-Montes and Coello (2005) ensembles four different constraint handling techniques: SF, SP, EC and SR. In this experiment, the algorithm is run 30 times for each benchmark function with a population size of 50. Constrained problems are generally considered more complex than the unconstrained problems and require a greater population size than that required by the unconstrained problems. Hence, a population size of 50 is considered in this experiment. For the performance of the algorithm, best solution, mean solution, median solution, worst solution and standard deviation are recorded for all the functions. The results for all 22 benchmark functions using TLBO with different constraint handling techniques are given in Table 7.

All the constraint handling techniques are compared based on the searching capability for the best and the mean solutions. It is observed from Table 7 that TLBO with ECHT produced superior results in searching for the best solution compared with the other constraint handling techniques for four benchmark functions: G14, G19, G21 and G23. TLBO with ECHT produced equivalent results to other constraint handling techniques for the remaining 18 benchmark functions. Moreover, TLBO with ECHT produced superior results for 11 benchmark functions in searching for the mean solution: G01, G02, G05, G07, G10, G14, G17, G18, G19, G21 and G23. For the rest of the benchmark functions, TLBO with ECHT produced equivalent results to the other constraint handling techniques. It is observed that all the constraint handling techniques except ECHT showed nearly equivalent performance in searching for the best as well as the mean solutions. It is also observed from the results that TLBO with ECHT performed 1.57 times better than other constraint handling techniques in searching for the best solutions and 2.2 times better in searching for the mean solutions.

Table 7. Results of constrained benchmark functions using teaching–learning-based optimization (TLBO) with different constraint handling techniques.

Function		SF-TLBO	SP-TLBO	EC-TLBO	SR-TLBO	ECHT-TLBO
G01 −15.00000						
	Best	−15.00000	−15.00000	−15.00000	−15.00000	−15.00000
	Mean	−14.99990	−14.99990	−14.99940	−14.99930	−15.00000
G02 −0.80362						
	Best	−0.80362	−0.80361	−0.80361	−0.80362	−0.80362
	Mean	−0.80237	−0.79723	−0.79865	−0.79548	−0.80359
G03 −1.00050						
	Best	−1.00050	−1.00050	−1.00050	−1.00050	−1.00050
	Mean	−1.00050	−1.00050	−1.00050	−1.00050	−1.00050
G04 −30665.53870						
	Best	−30665.53870	−30665.53870	−30665.53870	−30665.53870	−30665.53870
	Mean	−30665.53870	−30665.53870	−30665.53870	−30665.53870	−30665.53870
G05 5126.49670						
	Best	5126.49690	5126.49670	5126.49670	5126.49690	5126.49670
	Mean	5161.53880	5127.71820	5126.50580	5158.33170	5126.49670
G06 −6961.81390						
	Best	−6961.81390	−6961.81390	−6961.81390	−6961.81390	−6961.81390
	Mean	−6961.81390	−6961.81390	−6961.81390	−6961.81390	−6961.81390
G07 24.30620						
	Best	24.30620	24.30620	24.30630	24.30630	24.30620
	Mean	24.30830	24.30740	24.30920	24.30820	24.30630
G08 −0.09583						
	Best	−0.09583	−0.09583	−0.09583	−0.09583	−0.09583
	Mean	−0.09583	−0.09583	−0.09583	−0.09583	−0.09583
G09 680.63006						
	Best	680.63006	680.63006	680.63006	680.63006	680.63006
	Mean	680.63006	680.63006	680.63006	680.63006	680.63006
G10 7049.24800						
	Best	7049.24950	7049.24870	7049.24890	7049.24900	7049.24870
	Mean	7094.94359	7050.78570	7064.75955	7145.59676	7049.24880
G11 0.74990						
	Best	0.74990	0.74990	0.74990	0.74990	0.74990
	Mean	0.74990	0.74990	0.74990	0.74990	0.74990
G12 −1.00000						
	Best	−1.00000	−1.00000	−1.00000	−1.00000	−1.00000
	Mean	−1.00000	−1.00000	−1.00000	−1.00000	−1.00000
G13 0.05394						
	Best	0.05394	0.05394	0.05394	0.05394	0.05394
	Mean	0.05394	0.05394	0.05394	0.05394	0.05394
G14 −47.76490						
	Best	−47.23530	−47.75870	−47.65460	−47.76440	−47.76490
	Mean	−45.21888	−47.60297	−46.36839	−45.01175	−47.76480
G15 961.71502						
	Best	961.71502	961.71502	961.71502	961.71502	961.71502
	Mean	961.71502	961.71502	961.71502	961.71502	961.71502
G16 −1.90516						
	Best	−1.90516	−1.90516	−1.90516	−1.90516	−1.90516
	Mean	−1.90516	−1.90516	−1.90516	−1.90516	−1.90516
G17 8853.53970						
	Best	8853.87660	8862.34250	8853.53970	8853.88430	8853.53970
	Mean	8927.05966	8877.20021	8853.74350	8934.44236	8853.53970
G18 −0.86603						
	Best	−0.86595	−0.86600	−0.86596	−0.86600	−0.86603
	Mean	−0.86586	−0.86445	−0.86576	−0.86582	−0.86603
G19 32.65560						
	Best	32.89237	32.69850	32.80446	32.79010	32.65608
	Mean	33.27372	32.83084	33.21009	33.33337	32.65106
G21 193.72450						
	Best	193.73700	195.36281	193.72547	193.71960	193.72450
	Mean	225.37104	241.79494	234.72651	206.11650	193.73250

(*Continued*).

Table 7. Continued

Function		SF-TLBO	SP-TLBO	EC-TLBO	SR-TLBO	ECHT-TLBO
G23 −400.05510						
	Best	−380.36797	−324.67697	−385.13745	−371.23539	−396.95605
	Mean	−332.78161	−289.69716	−352.32407	−342.73427	−377.06648
G24 −5.50800						
	Best	−5.50800	−5.50800	−5.50800	−5.50800	−5.50800
	Mean	−5.50800	−5.50800	−5.50800	−5.50800	−5.50800

The performance of TLBO is also compared with other optimization methods with different constraint handling techniques, *i.e.* evolutionary strategies with stochastic ranking (ES + SR) (Runarsson and Yao 2000), simple multimembrane evolutionary strategy (SMES) (Mezura-Montes and Coello 2005), adaptive tradeoff model evolutionary strategy (ATMES) (Wang *et al.* 2008), multi-objective evolutionary strategy (Wang *et al.* 2007), improved stochastic ranking (ISR) (Runarsson and Yao 2005) and ensemble of constraint handling technique strategy (ECHT-EP) (Mallipeddi and Suganthan 2010). Comparison of the results for the performance of TLBO with other techniques is given in Table 8. It is observed from the results that the ECHT is an effective constraint handling method. ECHT outperformed all the other techniques in searching for the best and the mean solutions. ECHT with ES and TLBO produced similar results for the best solution and so both the algorithms performed equivalently to ECHT in searching for the best solution. In searching for the mean solution, ECHT with TLBO outperformed ES, by performing 1.22 times better.

3.3. *Experiment 3*

In this experiment, 13 constrained benchmark functions given in Mallipeddi and Suganthan (2010) are experimented with. For these 13 problems, all the constrained handling techniques mentioned in Experiment 2 are used with the TLBO. In this experiment, the population size is taken as 50. The results are compared based on the best solutions, median solutions, mean solutions, worst solutions and the standard deviation. The results of TLBO are compared with the results obtained using DE with all the constrained handling techniques presented by Mallipeddi and Suganthan (2010).

For this experiment, comparison is made for TLBO using all the constraint handling techniques. The results of the comparison are given in Table 9. It is observed from the results that ECHT with TLBO outperformed the other constraint handling techniques by performing 1.44, 1.3, 1.3 and 1.18 times better than SR, SF, SP and EC, respectively, in searching for the best solutions. Moreover, ECHT with TLBO performed 6, 6, 4 and 2.4 times better than SF, EC, SR and SP, respectively, in searching for the mean solution. The performance of TLBO is also compared with DE for all 13 problems. It is observed from the results that the performance of TLBO and DE with ECHT techniques is nearly the same and produced similar results for most of the benchmark functions except H02 and H09, for which both algorithms outperformed each other in searching for the mean solution.

4. Conclusions

All the nature-inspired algorithms, such as GA, PSO, ACO, ABC and HS, require algorithm parameters to be set for them to work properly. Proper selection of parameters is essential for the searching of the optimum solution by these algorithms. A change in the algorithm parameters

Table 8. Comparison of teaching–learning-based optimization (TLBO) with other optimization techniques for the constrained benchmark functions.

		(ES + SR)	SMES	ATMES	Multi–objective	ISR	ECHT–EP2	ECHT–TLBO
G01	Best	−15.0000	−15.0000	−15.0000	−15.0000	−15.0000	−15.0000	−15
	Mean	−15.0000	−15.0000	−15.0000	−15.0000	−15.0000	−15.0000	−15
G02	Best	−0.803515	−0.803601	−0.803339	−0.803550	−0.803619	−0.8036191	−0.80362
	Mean	−0.781975	−0.785238	−0.790148	−0.792610	−0.782715	−0.7998220	−0.80359
G03	Best	−1.000	−1.000	−1.000	−1.000	−1.001	−1.0005	−1.0005
	Mean	−1.000	−1.000	−1.000	−1.000	−1.001	−1.0005	−1.0005
G04	Best	−30665.5387	−30665.5387	−30665.5387	−30665.5387	−30665.5387	−30665.5387	−30665.5387
	Mean	−30665.5387	−30665.5387	−30665.5387	−30665.5387	−30665.5387	−30665.5387	−30665.5387
G05	Best	5126.497	5126.599	5126.4989	5126.4981	5126.497	5126.4967	5126.4967
	Mean	5128.881	5174.492	5127.648	5126.4981	5126.497	5126.4967	5126.4967
G06	Best	−6961.814	−6961.814	−6961.814	−6961.81388	−6961.814	−6961.8139	−6961.8139
	Mean	−6875.940	−6961.284	−6961.814	−6961.81388	−6961.814	−6961.8139	−6961.8139
G07	Best	24.307	24.327	24.306	24.30646	24.306	24.3062	24.3062
	Mean	24.374	24.475	24.316	24.3074	24.306	24.3063	24.3063
G08	Best	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.09582504	−0.09583
	Mean	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.09582504	−0.09583
G09	Best	680.63	680.632	680630	680.63006	680.63	680.63006	680.63006
	Mean	680.656	680.643	680.639	680.63006	680.63	680.63006	680.63006
G10	Best	7054.316	7051.903	7052.253	7049.2866	7049.248	7049.2483	7049.2487
	Mean	7559.192	7253.047	7250.437	7049.52544	7049.25	7049.249	7049.2488
G11	Best	0.75	0.75	0.75	0.75	0.75	0.7499	0.7499
	Mean	0.75	0.75	0.75	0.75	0.75	0.7499	0.7499
G12	Best	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1
	Mean	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1
G13	Best	0.05396	0.05399	0.05395	0.05395	0.05394	0.05394	0.05394
	Mean	0.06754	0.16639	0.05396	0.05395	0.06677	0.05394	0.05394

Table 9. Comparison of teaching–learning-based optimization (TLBO) with differential evolution (DE) for constrained benchmark functions H01–H13.

PROB		SF–TLBO	SP–TLBO	SR–TLBO	EC–TLBO	ECHT–TLBO	SF–DE	SP–DE	SR–DE	EC–DE	ECHT–DE
H01	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.59E−11	1.24E−85	7.57E−81	5.59E−11	8.29E−83
	Mean	5.67E+03	0.00E+00	0.00E+00	5.78E+03	0.00E+00	4.58E+03	2.95E−83	1.37E−74	4.58E+03	2.66E−78
H02	Best	−9.56E−01	−2.27E+00	5.36E−01	−9.18E−01	−2.28E+00	−1.01E+00	−2.27E+00	7.52E−02	−1.01E+00	−2.28E+00
	Mean	8.72E−01	−8.80E−01	5.95E+00	9.72E−01	−2.23E+00	7.73E−01	−1.42E+00	2.17E+00	7.73E−01	−2.25E+00
H03	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.08E−81	2.95E−85	6.25E−80	1.08E−81	1.19E−83
	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.05E−78	8.31E−83	2.20E−77	2.05E−78	6.90E−81
H04	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.89E−93	4.17E−94	2.50E−94	3.89E−93	4.91E−95
	Mean	0.00E+00	0.00E+00	0.00E+00	1.72E+02	0.00E+00	1.12E+03	5.81E−92	3.41E−91	1.12E+03	1.01E−92
H05	Best	−2.01E+01	−2.01E+01	−1.91E+01	−2.01E+01	−2.01E+01	−2.01E+01	−2.01E+01	−1.90E+01	−2.01E+01	−2.01E+01
	Mean	−1.98E+01	−1.99E+01	−1.52E+01	−1.98E+01	−2.01E+01	−1.95E+01	−1.99E+01	−1.67E+01	−1.94E+01	−2.01E+01
H06	Best	−8.38E+00	−8.38E+00	−8.38E+00	−8.38E+00	−8.38E+00	−8.38E+00	−8.38E+00	−8.38E+00	−8.38E+00	−8.38E+00
	Mean	−2.41E+00	−7.65E+00	−8.04E+00	−1.21E+00	−8.38E+00	−4.10E+00	−8.19E+00	−8.17E+00	−2.98E+00	−8.38E+00
H07	Best	−7.62E+00	−7.62E+00	−7.62E+00	−7.62E+00	−7.62E+00	−7.62E+00	−7.62E+00	−7.62E+00	−6.02E+00	−7.62E+00
	Mean	−6.99E+00	−6.83E+00	−6.42E+00	−4.48E+00	−7.62E+00	−4.50E+00	−6.33E+00	−6.82E+00	−1.86E+00	−7.62E+00
H08	Best	−1.55E+02	−4.80E+02	−4.39E+02	1.01E+03	−4.84E+02	−1.62E+02	−4.82E+02	−4.72E+02	5.00E+02	−4.84E+02
	Mean	2.69E+02	−4.40E+02	−3.98E+02	6.22E+02	−4.84E+02	1.80E+02	−4.80E+02	−4.67E+02	5.00E+02	−4.84E+02
H09	Best	−6.84E+01	−4.50E+01	−5.12E+01	−6.84E+01	−6.84E+01	−6.84E+01	−5.15E+01	−5.94E+01	−6.84E+01	−6.84E+01
	Mean	−6.88E+01	−3.70E+01	−5.03E+01	−6.92E+01	−6.81E+01	−6.75E+01	−4.47E+01	−5.04E+01	−6.76E+01	−6.79E+01
H10	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Mean	4.05E+00	3.76E+00	5.63E+00	3.30E+00	5.84E−02	4.20E+00	3.90E+00	5.39E+00	4.20E+00	5.99E−01
H11	Best	5.81E+02	5.81E+02	5.81E+02	5.81E+02	5.81E+02	5.81E+02	5.81E+02	8.94E+02	5.81E+02	5.81E+02
	Mean	6.18E+02	5.81E+02	1.01E+03	6.01E+02	5.81E+02	6.01E+02	5.81E+02	9.74E+02	6.01E+02	5.81E+02
H12	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.14E−29	1.25E−30	3.18E+01	3.20E−29	1.54E−32
	Mean	8.57E+00	0.00E+00	7.92E+00	8.19E+00	0.00E+00	3.99E+01	7.97E−02	4.80E+01	4.28E+01	4.55E−31
H13	Best	−4.64E+01	−4.64E+01	−4.64E+01	−4.64E+01	−4.64E+01	−4.64E+01	−4.64E+01	−4.64E+01	−4.64E+01	−4.64E+01
	Mean	−4.63E+01	−4.64E+01	−4.60E+01	−4.63E+01	−4.64E+01	−4.63E+01	−4.63E+01	−4.63E+01	−4.61E+01	−4.64E+01

influences the effectiveness of the algorithm. To avoid this difficulty, an optimization method, TLBO, which is algorithm parameter free, is presented in this article. This method works on the effect of the influence of a teacher on learners. Like other nature-inspired algorithms, TLBO is a population-based method which uses a population of solutions to proceed to the global solution. 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 phases, and ABC, which uses employed, onlooker and scout bees phases, TLBO uses two different phases, the teacher phase and the learner phase. TLBO uses the mean value of the population to update the solution. TLBO implements greediness to accept the good solution, like ABC. In the teacher phase of TLBO, the update of a solution from an old solution is considered as the exploration and the greedy selection which follows it is considered as the exploitation. Similarly, in the learner phase, the updating of the solution is the exploration and the greedy selection is the exploitation. So, TLBO incorporates both exploration and exploitation effectively in a balanced manner.

In this article, TLBO is experimented with on 25 unconstrained benchmark functions defined in CEC 2005, 22 constrained benchmark function defined in CEC 2006 and 13 other constrained benchmark functions. Thus, the TLBO algorithm is experimented with on 60 different benchmark functions having different characteristics. For the constrained benchmark functions, the TLBO algorithm is experimented with using five different constraint handling techniques. The performance of the TLBO algorithm is compared with that of other optimization methods. Results have shown the satisfactory performance of the TLBO algorithm for the constrained as well as unconstrained benchmark functions.

TLBO algorithm shows a better performance with less computational effort for large-scale problems, *i.e.* problems of a high dimensionality. This algorithm can be used for the optimization of engineering design and manufacturing applications. However, as the algorithm uses the greedy selection procedure at the end of both the teacher and the learner phases, it may sometimes lead to an increase in computational effort. The algorithm may be made more efficient by modifying the greedy selection at the end of both phases. Consideration of different teachers for different subjects is another aspect of future research work.

References

Ahrari, A.A. and Atai, A., 2010. Grenade explosion method—a novel tool for optimization of multimodal functions. *Applied Soft Computing*, 10, 1132–1140.

Akay, B. and Karaboga, D., 2010. A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences* (in press), doi:10.1016/j.ins.2010.07.015.

Auger, A. and Hansen, N., 2005. A restart CMA evolution strategy with increasing population size. *In*: *2005 IEEE congress on evolutionary computation*, 2–4 September Edinburgh, Vol. 2. Piscataway, NJ: IEEE Press, 1769–1776.

Ballester, P.J., *et al.*, 2005. Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX. *In*: *2005 IEEE congress on evolutionary computation*, 2–4 September Edinburgh, Vol. 1. Piscataway, NJ: IEEE Press, 498–505.

Das, S., *et al.*, 2011. Real-parameter evolutionary multimodal optimization—a survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 1, 71–88.

Deb, K., 2000. An efficient constraint handling method for genetic algorithms. *Computer Methods, Applied Mechanics and Engineering* 186, 311–338.

Dorigo, M., 1992. *Optimization, learning and natural algorithms*. Thesis (PhD). Politecnico di Milano.

Eusuff, M. and Lansey, E., 2003. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 129, 210–225.

Farmer, J.D., Packard, N., and Perelson, A., 1986. The immune system, adaptation and machine learning. *Physica D*, 22, 187–204.

Geem, Z.W., Kim, J.H., and Loganathan, G.V., 2001. A new heuristic optimization algorithm: harmony search. *Simulation*, 76, 60–65.

Holland, J., 1975. *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Jian, M.-C., 2006. *Introducing recombination with dynamic linkage discovery to particle swarm optimization.* Department of Computer Science, National Chiao Tung University, Taiwan, Technical Report NCL-TR-2006006.

Karaboga, D., 2005. *An idea based on honey bee swarm for numerical optimization.* Computer Engineering Department, Erciyes University, Turkey, Technical Report-TR06.
Kennedy, J. and Eberhart, R.C., 1995. Particle swarm optimization. *In*: *Proceedings of IEEE international conference on neural networks*, 27 November–1 December Perth. Piscataway, NJ: IEEE Press, 1942–1948.
Liang, J.J. and Suganthan, P.N., 2005. Dynamic multi-swarm particle swarm optimizer with local search. *In*: *2005 IEEE congress on evolutionary computation*, 2–4 September Edinburgh, Vol. 1. Piscataway, NJ: IEEE Press, 522–528.
Liang, J.J., *et al.*, 2006. Problem definitions and evolution criteria for the CEC 2006 special session on constrained real-parameter optimization. Nanyang Technological University, Singapore, Technical Report. Available from: http://www.ntu.edu.sg/home/EPNSugan
Mallipeddi, R. and Suganthan, P.N., 2010. Ensemble of constraint handling techniques. *IEEE Transactions on Evolutionary Computation*, 14, 561–579.
Mezura-Montes, E. and Coello, C.A.C., 2005. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 9, 1–17.
Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22, 52–67.
Qin, A.K. and Suganthan, P.N., 2005. Self-adaptive differential evolution algorithm for numerical optimization. *In*: *2005 IEEE congress on evolutionary computation*, 2–4 September Edinburgh, Vol. 2. Piscataway, NJ: IEEE Press, 1785–1791.
Rao, R.V., Savsani, V.J., and Vakharia, D.P., 2011. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43, 303–315.
Rao, R.V., Savsani, V.J., and Vakharia, D.P., 2012. Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information Sciences*, 183, 1–15.
Rashedi, E., Nezamabadi-pour, H., and Saryazdi, S., 2009. GSA: a gravitational search algorithm. *Information Sciences*, 179, 2232–2248.
Ronkkonen, J., Kukkonen, S., and Price, K.V., 2005. Real-parameter optimization with differential evolution. *In*: *2005 IEEE congress on evolutionary computation*, 2–4 September Edinburgh, Vol. 1. Piscataway, NJ: IEEE Press, 506–513.
Runarsson, T.P. and Yao, X., 2000. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4, 284–294.
Runarsson, T.P. and Yao, X., 2005. Search biases in constrained evolutionary optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 35, 233–243.
Simon, D., 2008. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, 12, 702–713.
Storn, R. and Price, K., 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341–359.
Suganthan, P.N., *et al.*, 2005. *Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization.* Nanyang Technological University, Singapore, Technical Report. Available from: http://www.ntu.edu.sg/home/EPNSugan
Takahama, T. and Sakai, S., 2006. Constrained optimization by the constrained differential evolution with gradient-based mutation and feasible elites. *In*: *Proceedings of IEEE congress on evolutionary computation*, 16–21 July Vancouver. Piscataway, NJ: IEEE Press, 1–8.
Tessema, B. and Yen, G.G., 2006. A self-adaptive penalty function based algorithm for constrained optimization. *In*: *Proceedings of IEEE congress on evolutionary computation*, 16–21 July Vancouver. Piscataway, NJ: IEEE Press, 246–253.
Wang, Y., *et al.*, 2007. Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. *IEEE Transactions on Systems, Man and Cybernetics*, 37, 560–575.
Wang, Y., *et al.*, 2008. An adaptive tradeoff model for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 12, 80–92.
Zhou, A., *et al.*, 2011. Multiobjective evolutionary algorithms: a survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 1, 32–49.