Decision Science Letters 2 (2012) 23–34

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Improved teaching learning based optimization for global function optimization

Suresh Chandra Satapathy[a] and Anima Naik[b*]

[a]Sr Member IEEE, ANITS,Vishakapatnam, India
[b]MITS, India

C H R O N I C L E

Article history:
Received July 25, 2012
Accepted October 10, 2012
Available online
October 19 2012

Keywords:
Optimization technique
TLBO
Convergence
Performance

A B S T R A C T

Teaching–Learning-Based Optimization (TLBO) is recently being used as a new, reliable, accurate and robust optimization technique scheme for global optimization over continuous spaces. This paper presents an improved variant of TLBO algorithm, called Improved Teaching–Learning-Based Optimization (ITLBO). A performance comparison of the proposed method is provided against the original TLBO and some other algorithms. The improved TLBO algorithm shows a marked improvement in performance over the traditional TLBO on several benchmark optimization problems.

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.

Rao et al. (2011, 2012) proposed a teaching-learning based optimization (TLBO) algorithm based on the natural phenomenon of teaching and learning. The implementation of TLBO does not require the determination of any algorithm specific controlling parameters, which makes the algorithm robust and powerful. TLBO requires only common controlling parameters like population size and number of generations for its working. In this way, TLBO can be said as an algorithm specific parameter-less algorithm. Rao and Patel (2012) investigated the performance of TLBO algorithm for different elite sizes, population sizes and number of generations considering various constrained bench mark problems available in the literature to identify their effect on the exploration and exploitation capacity of the algorithm. Satapathy and Naik (2011) tried to propose a new approach to use TLBO to cluster data. They have proved that the TLBO algorithm can be used to cluster arbitrary data. Again they

* Corresponding author.
E-mail addresses: animanaik@gmail.com (A. Naik)

doi: 10.5267/j.dsl.2012.10.005

have shown another good characteristic of TLBO algorithm, that the cost of computations are less in handling high dimensional problems in compare to other algorithms (Chandra Satapathy et al., 2012). TLBO also has been applied to the area of computer network by optimizing the multicast tree (Naik et al., 2012). It is capable of handling power system problem easily where optimal solution of the unit maintenance scheduling problem in which the cost reduction is as important as reliability (Satapathy et al., 2012). TLBO algorithm has been adapted to solve multi objective problems of an economic load dispatch problem with incommensurable objectives (Krishnanand et al., 2011). From literature, we can see while clustering the data using fuzzy c-means (FCM) and hard c-means (HCM), the sensitivity to tune the initial clusters centers have captured the attention of the clustering communities for quite a long time. This problem has been addressed by TLBO in (Naik et al., 2012). In the area of feature selection, also TLBO is showing its performance in connection with Rough set theory. Empirical results reveal that the Rough TLBO approach could perform better in terms of finding optimal features and doing so in quick time in comparison with GA, PSO and DE (Naik et al., 2012). Again there is some modification of TLBO have been done to improve its performance in optimization for global function optimization (Chandra Satapathy et al., 2012). There are good numbers of applications of TLBO from different papers (Rao & Patel, 2012; Rao & Savsani, 2012; Vedat Toğan, 2012; Rao & Kalyankar, 2012). Looking into the diverse applications of TLBO and effectiveness of TLBO, in this paper, we present yet another modification on TLBO to improve its performance in terms of convergence speed while solving unconstrained optimization problem. Our proposed improved TLBO (ITLBO) is compared with various other algorithms like PSO, DE, TLBO, OEA, HPSO-TVAC, CLPSO and APSO on several benchmark functions are the performance characteristics are provided to show that ITLBO performs better than all other compared techniques.

The remainder of this paper is organized as follows. The conventional TLBO is explained in detail in Sections 2, ITLBO in section 3. Numerical Experiment and results demonstrating the performance of ITLBO in comparison with other optimization algorithm are presented in Section 4. Section 5 concludes this paper.

2. Teaching–learning-based optimization

This optimization method is based on the effect of the influence of a teacher on the output of learners in a class. It is a population based method and like other population based methods, it uses a population of solutions to proceed to the global solution. A group of learners constitutes the population in TLBO. In any optimization algorithms, there are numbers of different design variables. Different design variables in TLBO are analogous to various subjects offered to learners and the learners' result is analogous to the 'fitness', as in other population-based optimization techniques. As the teacher is considered the most learned person in the society, the best solution so far is analogous to Teacher in TLBO. The process of TLBO is divided into two parts. The first part consists of the 'Teacher Phase' and the second part consists of the 'Learner Phase'. The 'Teacher Phase' means learning from the teacher and the 'Learner Phase' means learning through the interaction between learners. In the sub-sections below, we briefly discuss the implementation of TLBO.

A Initialization

Following are the notations used for describing the TLBO:
N: number of learners in a class i. e. "class size"
D: number of courses offered to the learners
$MAXIT$: maximum number of allowable iterations

The population X is randomly initialized by a search space bounded by matrix of N rows and D columns. The jth parameter of the ith learner is assigned values randomly using the equation

$$x_{(i,j)}^{0} = x_j^{min} + rand \times (x_j^{max} - x_j^{min}) \tag{1}$$

where $rand$ represents a uniformly distributed random variable within the range (0, 1), x_j^{min} and x_j^{max} represent the minimum and maximum value for jth parameter. The parameters of ith learner for the generation g are given by

$$X_{(i)}^{g} = [x_{(i,1)}^{g}, x_{(i,2)}^{g}, x_{(i,3)}^{g}, \dots\dots, x_{(i,j)}^{g}, \dots\dots, x_{(i,D)}^{g}] \tag{2}$$

B Teacher Phase

The mean parameter M^g of each subject of the learners in the class at generation g is given as

$$M^g = \left[m_1^g, m_2^g, \dots\dots, m_j^g, \dots\dots, m_D^g\right]. \tag{3}$$

The learner with the minimum objective function value is considered as the teacher $X_{Teacher}^{g}$ for respective iteration. The Teacher phase makes the algorithm proceed by shifting the mean of the learners towards its teacher. To obtain a new set of improved learners a random weighted differential vector is formed from the current mean and the desired mean parameters and added to the existing population of learners.

$$Xnew_{(i)}^{g} = X_{(i)}^{g} + rand \times (X_{Teacher}^{g} - T_F M^g) \tag{4}$$

T_F is the teaching factor, which decides the value of mean to be changed. Value of T_F can be either 1 or 2. The value of T_F is decided randomly with equal probability as,

$$T_F = round[1 + rand(0,1)\{2 - 1\}], \tag{5}$$

where 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 Eq. (5). After conducting a number of experiments on many benchmark functions it is concluded that the algorithm performs better if the value of T_F is between 1 and 2. However, the algorithm is found to perform much better if the value of T_F is either 1 or 2 and hence to simplify the algorithm, the teaching factor is suggested to take either 1 or 2 depending on the rounding up criteria given by Eq.(5). If $Xnew_{(i)}^{g}$ is found to be a superior learner than $X_{(i)}^{g}$ in generation g , than it replaces inferior learner $X_{(i)}^{g}$ in the matrix.

C Learner Phase

In this phase, the interaction of learners with one another takes place. The process of mutual interaction tends to increase the knowledge of the learner. The random interaction among learners improves his or her knowledge. For a given learner $X_{(i)}^{g}$, another learner $X_{(r)}^{g}$ is randomly selected$(i \neq r)$. The ith parameter of the matrix $Xnew$ in the learner phase is given as

$$Xnew_{(i)}^{g} = \begin{cases} X_{(i)}^{g} + rand \times \left(X_{(i)}^{g} - X_{(r)}^{g}\right) & if\ f\left(X_{(i)}^{g}\right) < f(X_{(r)}^{g}) \\ X_{(i)}^{g} + rand \times \left(X_{(r)}^{g} - X_{(i)}^{g}\right) & oterwise \end{cases} \tag{6}$$

D Algorithm Termination

The algorithm is terminated after $MAXIT$ iterations are completed. Details of TLBO can be refereed in (Rao et al., 2011).

3. Improved Teaching–Learning-Based Optimizer (ITLBO)

In the traditional TLBO (Rao et al., 2011), the teacher phase makes the algorithm proceed by shifting the mean of the learners towards its teacher.

To obtain a new set of improved learners a *random weighted differential vector* is formed from the current mean and the desired mean parameters and added to the existing population of learners. Similarly, in the learner phase the algorithm proceeds by random interaction among learners improve his or her knowledge. To obtain a new set of improved learners a random weighted differential vector is formed from a given learner$X_{(i)}^{g}$, another learner $X_{(r)}^{g}$ is randomly selected$(i \neq r)$ and added to the existing learner. In our proposed algorithm we propose to vary this random weighed differential vector in a random manner in the range (0.5, 1) by using the relation

0.5*(1+rand (0, 1)), (7)

where rand (0, 1) is a uniformly distributed random number within the range [0, 1]. Therefore, the mean value of this weighted differential scale factor is 0.75. This allows for stochastic variations in the amplification of the difference vector and thus helps retain population diversity as the search progresses. Even when the tips of most of the population vectors point to locations clustered near a local optimum due to the randomly scaled difference vector, a new trial vector has fair chances of pointing at an even better location on the multimodal functional surface. Therefore, the fitness of the best vector in a population is much less likely to get stagnant until a truly global optimum is reached. So the new set of improved learners can be made by using equation in the teacher phase

$$Xnew_{(i)}^{g} = X_{(i)}^{g} + 0.5 * \left(1 + \mathrm{rand}(0,1)\right) * (X_{Teacher}^{g} - T_F M^{g}) \tag{8}$$

So the new set of improved learners can be made by using equation in the learner phase

$$Xnew_{(i)}^{g} = \begin{cases} X_{(i)}^{g} + 0.5 * \left(1 + \mathrm{rand}(0,1)\right) * \left(X_{(i)}^{g} - X_{(r)}^{g}\right) & if\ f\left(X_{(i)}^{g}\right) < f(X_{(r)}^{g}) \\ X_{(i)}^{g} + 0.5 * \left(1 + \mathrm{rand}(0,1)\right) * \left(X_{(r)}^{g} - X_{(i)}^{g}\right) & oterwise \end{cases} \tag{9}$$

4. Numerical experiments and results

4.1 Experiments 1: ITLBO vs. PSO, DE and TLBO

4.1.1 Settings

For all experiments performed in this section, the values of the common parameters used in each algorithm such as population size and total evaluation number are chosen to be the same. Population size is 20 and the maximum number fitness function evaluation is 40,000 for all functions. The other specific parameters of algorithms are given below:

PSO Settings: Cognitive and social components c_1,c_2 are constants that can be used to change the weighting between personal and population experience, respectively. In our experiments cognitive and social components were both set to 2. Inertia weight, which determines how the previous velocity of the particle influences the velocity in the next iteration, was 0.5.

DE Settings: In DE, F is a real constant which affects the differential variation between two Solutions and set to F = 0.5*(1+ rand (0, 1)) where rand (0, 1) is a uniformly distributed random number within the range [0, 1] in our experiments. Value of crossover rate, which controls the change of the diversity of the population, was chosen to be R = ($R_{max} - R_{min}$) * ($MAXIT$–$iter$) / $MAXIT$ where $R_{max} = 1$ and R_{min}=0.5 are the maximum and minimum values of scale factor R, $iter$ is the current iteration number and $MAXIT$ is the maximum number of allowable iterations as recommended in (Swagatam Das et al., 2008).

TLBO Settings: For TLBO there is no such constant to set.

ITLBO Settings: For ITLBO there is no such constant to set.

4.1.2 Benchmark functions

We used 20 benchmark problems in order to test the performance of the PSO, DE, TLBO and the ITLBO algorithms. This set is large enough to include many different kinds of problems such as unimodal, multimodal, regular, irregular, separable, non-separable and multidimensional. Initial range, formulation, characteristics and the dimensions of these problems are listed in Table 1.

Table 1

Benchmark functions used in experiments 1. D: Dimension, C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable

No.	Function	D	C	Range	Formulation	Value
f_1	Step	30	US	[-100,100]	$f(x)=\sum_{i=1}^{D}(\lfloor x_i+0.5\rfloor)^2$	$f_{min}=0$
f_2	Sphere	30	US	[-100,100]	$f(x)=\sum_{i=1}^{D}x_i^2$	$f_{min}=0$
f_3	SumSquares	30	US	[-100,100]	$f(x)=\sum_{i=1}^{D}ix_i^2$	$f_{min}=0$
f_4	Quartic	30	US	[-1.28,1.28]	$f(x)=\sum_{i=1}^{D}ix_i^4+random(0,1)$	$f_{min}=0$
f_5	Zakharov	10	UN	[-5,10]	$f(x)=\sum_{i=1}^{D}x_i^2+(\sum_{i=1}^{D}0.5ix_i)^2+(\sum_{i=1}^{D}0.5ix_i)^4$	$f_{min}=0$
f_6	Schwefel 1.2	30	UN	[-100,100]	$f(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2$	$f_{min}=0$
f_7	Schwefel 2.22	30	UN	[-10,10]	$f(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	$f_{min}=0$
f_8	Schwefel 2.21	30	UN	[-100,100]	$f(x)=\max_i\{\lvert x_i\rvert,1\le i\le D\}$	$f_{min}=0$
f_9	Bohachevsky1	2	MS	[-100,100]	$f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$	$f_{min}=0$
f_{10}	Bohachevsky2	2	MS	[-100,100]	$f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)*\cos(4\pi x_2)+0.3$	$f_{min}=0$
f_{11}	Bohachevsky3	2	MS	[-100,100]	$f(x)=x_1^2+2x_2^2-0.3\cos((3\pi x_1)+(4\pi x_2))+0.3$	$f_{min}=0$
f_{12}	Booth	2	MS	[-10,10]	$f(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	$f_{min}=0$
f_{13}	Rastrigin	30	MS	[-5.12,5.12]	$f(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$	$f_{min}=0$
f_{14}	Schaffer	2	MN	[-100,100]	$f(x)=\frac{sin^2\left(\sqrt{x_1^2+x_2^2}\right)-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	$f_{min}=0$
f_{15}	Six Hump Camel Back	2	MN	[-5,5]	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	$f_{min}=-1.03163$
f_{16}	Griewank	30	MN	[-600,600]	$f(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	$f_{min}=0$
f_{17}	Ackley	30	MN	[-32,32]	$f(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{D}\cos(2*pi*x_i)\right)+20+e$	$f_{min}=0$
f_{18}	Multimod	30		[-10,10]	$f(x)=\sum_{i=1}^{D}\lvert x_i\rvert\prod_{i=1}^{D}\lvert x_i\rvert$	$f_{min}=0$
f_{19}	Noncontinuous Rastrigin	30	MS	[-5.12,5.12]	$f(x)=\sum_{i=1}^{D}[y_i^2-10\cos(2\pi y_i)+10]$ Where $y_i=\begin{cases}x_i & \lvert x_i\rvert<0.5\\ \frac{round(2x_i)}{2} & \lvert x_i\rvert\ge 0.5\end{cases}$	$f_{min}=0$
f_{20}	Weierstrass	30		[-0.5, 0.5]	$f(x)=\sum_{i=1}^{D}(\sum_{k=0}^{kmax}[a^k\cos(2\pi b^k(x_i+0.5))])-D\sum_{k=0}^{kmax}[a^k\cos(2\pi b^k(x_i+0.5))], where\ a=0.5, b=3, kmax=20$	$f_{min}=0$

Table 2

Performance comparisons of PSO, DE, TLBO, ITLBO

No.	Function	Global min/max		PSO	DE	TLBO	ITLBO
f_1	Step	$f_{min} = 0$	Mean	203.3667	0	0	0
			Std	56.2296	0	0	0
f_2	Sphere	$f_{min} = 0$	Mean	6.1515e-09	7.2140e-14	1.0425e-281	0
			Std	7.6615e-10	5.8941e-14	0	0
f_3	SumSquares	$f_{min} = 0$	Mean	3.7584e-14	6.1535e-15	1.5997e-281	0
			Std	1.0019e-14	3.0555e-15	0	0
f_4	Quartic	$f_{min} = 0$	Mean	1.9275	0.0253	2.3477e-04	1.5209e-04
			Std	1.4029	0.0075	1.7875e-04	1.1235e-04
f_5	Zakharov	$f_{min} = 0$	Mean	141.0112	66.8339	1.4515e-281	0
			Std	40.7567	14.4046	0	0
f_6	Schwefel 1.2	$f_{min} = 0$	Mean	9.3619e-08	5.3494e-13	2.6061e-270	0
			Std	6.6112e-08	4.6007e-13	0	0
f_7	Schwefel 2.22	$f_{min} = 0$	Mean	9.3293	3.9546e-07	3.1583e-137	1.0079e-238
			Std	3.6619	1.9283e-07	1.7188e-137	0
f_8	Schwefel 2.21	$f_{min} = 0$	Mean	60.9603	1.5340	4.3819e-136	1.1377e-226
			Std	4.0761	0.3900	1.5668e-136	0
f_9	Bohachevsky1	$f_{min} = 0$	Mean	0	0	0	0
			Std	0	0	0	0
f_{10}	Bohachevsky2	$f_{min} = 0$	Mean	0	0	0	0
			Std	0	0	0	0
f_{11}	Bohachevsky3	$f_{min} = 0$	Mean	0	0	0	0
			Std	0	0	0	0
f_{12}	Booth	$f_{min} = 0$	Mean	0	0	0	0
			Std	0	0	0	0
f_{13}	Rastrigin	$f_{min} = 0$	Mean	76.2918	5.6344	0	0
			Std	17.1005	1.8667	0	0
f_{14}	Schaffer	$f_{min} = 0$	Mean	0.0097	0.0029	0.0066	0
			Std	0.0025	0.0011	0.0045	0
f_{15}	Six Hump Camel Back	$f_{min} = -1.03163$	Mean	-1.0316	-1.0316	-1.0316	-1.0316
			Std	0	0	0	0
f_{16}	Griewank	$f_{min} = 0$	Mean	7.6291e-08	5.7841e-011	0	0
			Std	4.0012e-09	1.6914e-011	0	0
f_{17}	Ackley	$f_{min} = 0$	Mean	14.0614	7.3814e-08	1.7171e-15	1.7702e-15
			Std	2.0125	3.0453e-08	1.5979e-15	1.2434e-15
f_{18}	Multimod	$f_{min} = 0$	Mean	2.1994e-257	2.5678e-255	0	0
			Std	0	0	0	0
f_{19}	Noncontinuous Rastrigin	$f_{min} = 0$	Mean	100.3984	13.9237	0	0
			Std	28.7062	2.3146	0	0
f_{20}	Weierstrass	$f_{min} = 0$	Mean	12.0447	1.5388e-05	0	0
			Std	2.6160	1.0139e-05	0	0

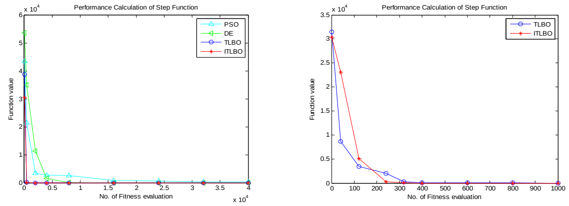
Table 3
No. of fitness evaluation comparisons of PSO, DE, TLBO, ITLBO

No.	Function		PSO	DE	TLBO	ITLBO
f_1	Step	Mean	40,000	2.4833e+4 (≈24833)	712	413.3333(≈413)
		Std	0	753.6577 (≈754)	30.4450(≈31)	19.1237(≈20)
f_2	Sphere	Mean	40,000	40,000	40,000	2.6651e+04(≈26651)
		Std	0	0	0	194.5274 (≈195)
f_3	SumSquares	Mean	40,000	40,000	40,000	27315
		Std	0	0	0	114.5502 (≈115)
f_4	Quartic	Mean	40,000	40,000	40,000	40,000
		Std	0	0	0	0
f_5	Zakharov	Mean	40,000	40,000	40,000	2.7022e+04 (≈27022)
		Std	0	0	0	117.0580 (≈118)
f_6	Schwefel 1.2	Mean	40,000	40,000	40,000	28100
		Std	0	0	0	83.5126 (≈84)
f_7	Schwefel 2.22	Mean	40,000	40,000	40,000	40,000
		Std	0	0	0	0
f_8	Schwefel 2.21	Mean	40,000	40,000	40,000	40,000
		Std	0	0	0	0
f_9	Bohachev sky1	Mean	3200	4.1111e+03 (≈4111)	1940	1200
		Std	51.6398 (≈52)	117.5409(≈118)	79.8308(≈78)	36.2329(≈37)
f_{10}	Bohachev sky2	Mean	3.1429e+03(≈3142)	4.2844e+003(≈4284)	2.0836e+03(≈2083)	1176
		Std	200.5150 (≈201)	201.8832 (≈202)	140.3219(≈141)	37.1276(≈38)
f_{11}	Bohachev sky3	Mean	4945	7.7822e+03(≈7782)	2148	1179
		Std	168.1727(≈169)	140.2739(≈141)	51.4009(≈52)	32.2931(≈33)
f_{12}	Booth	Mean	6420	1.2554e+004(≈12554)	3.4277e+03(≈3427)	2.3086e+03(≈2308)
		Std	18.3935(≈19)	803.3543(≈804)	121.4487(≈122)	145.5112(≈146)
f_{13}	Rastrigin	Mean	40,000	40,000	4.4533e+03(≈4453)	2.0688e+03(≈2068)
		Std	0	0	544.6047(≈545)	57.1781(≈58)
f_{14}	Schaffer	Mean	40,000	40,000	40,000	4.0267e+03(≈4026)
		Std	0	0	0	145.6414(≈146)
f_{15}	Six Hump Camel Back	Mean	800	1.5556e+03(≈1555)	720	417.1429(≈418)
		Std	99.2278(≈100)	136.7738(≈137)	33.0289(≈34)	20.0348
f_{16}	Griewank	Mean	40,000	40,000	2916	1.7455e+03(≈1745)
		Std	0	0	145.0686(≈146)	52.5381(≈53)
f_{17}	Ackley	Mean	40,000	40,000	40,000	40,000
		Std	0	0	0	0
f_{18}	Multimod	Mean	40,000	40,000	3488	2132
		Std	0	0	30.2715(≈31)	72.3935(≈73)
f_{19}	Noncontinuous Rastrigin	Mean	40,000	40,000	6.1891e+03(≈6189)	2.2235e+03(≈2223)
		Std	0	0	75.6887(≈76)	31.8082(≈32)
f_{20}	Weierstrass	Mean	40,000	40,000	4.0178e+03(≈4017)	2.4667e+03(≈2466)
		Std	0	0	110.5696(≈111)	100.9870(≈101)

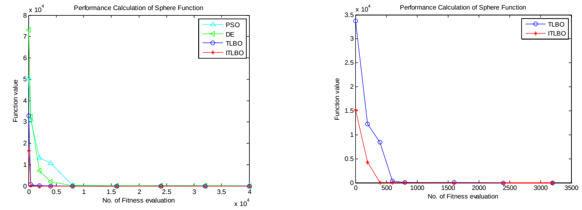
(values in brackets in the table are the absolute values of FEs)

Table 4
t value , significant at a 0.05 level of significance by two tailed test using table 2.

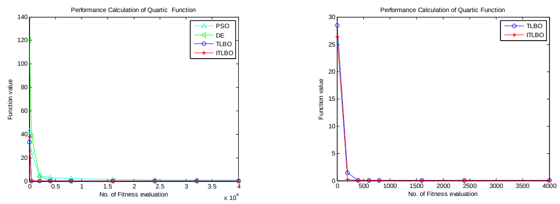
Function No.	PSO/ITLBO	DE/ITLBO	TLBO/ITLBO	Function No.	PSO/ITLBO	DE/ITLBO	TLBO/ITLBO
f_1	+	NA	NA	f_{11}	NA	NA	NA
f_2	+	+	+	f_{12}	NA	NA	NA
f_3	+	+	+	f_{13}	+	+	NA
f_4	+	+	.	f_{14}	+	+	+
f_5	+	+	+	f_{15}	NA	NA	NA
f_6	+	+	+	f_{16}	+	+	NA
f_7	+	+	+	f_{17}	+	+	+
f_8	+	+	+	f_{18}	+	+	NA
f_9	NA	NA	NA	f_{19}	+	+	NA
f_{10}	NA	NA	NA	f_{20}	+	+	NA



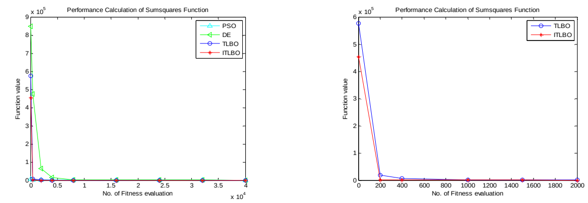
f_1



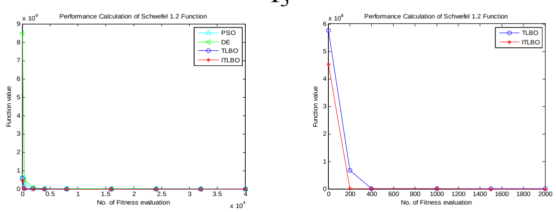
f_2



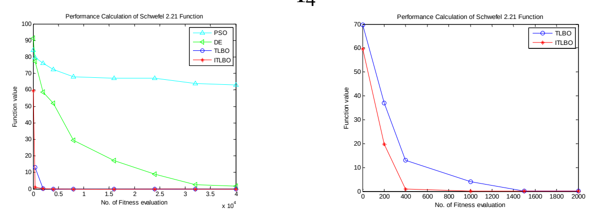
f_3



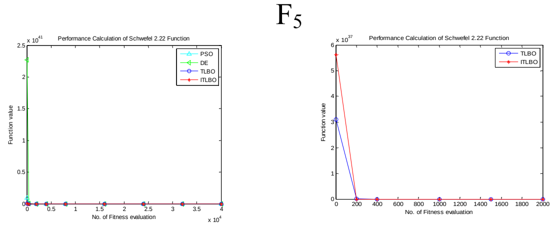
f_4



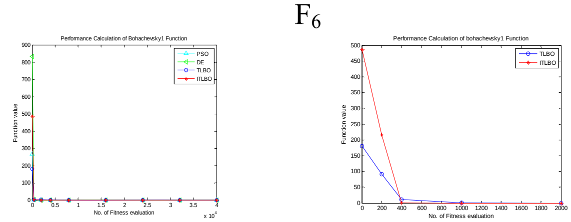
F_5



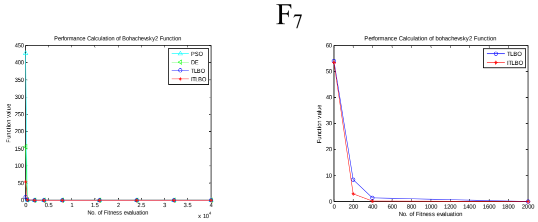
F_6



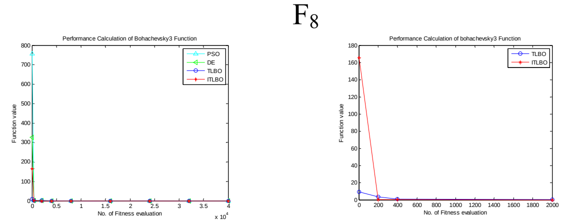
F_7



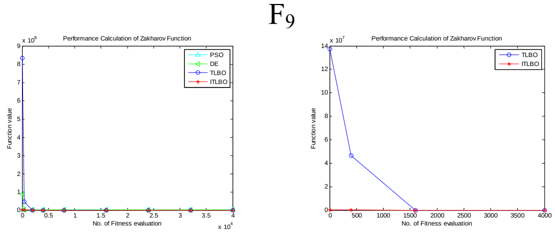
F_8



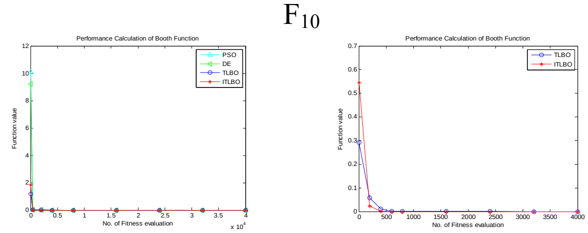
F_9



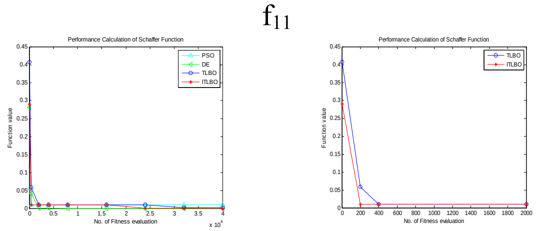
F_{10}



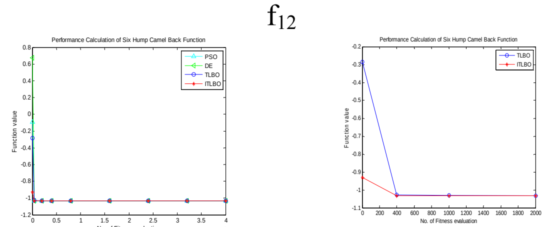
f_{11}



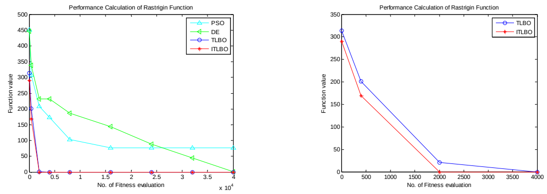
f_{12}



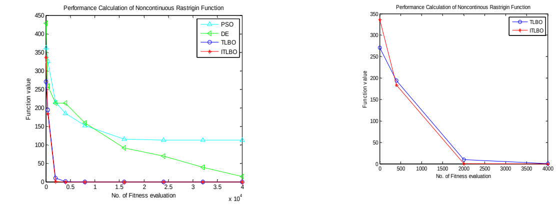
f_{13}



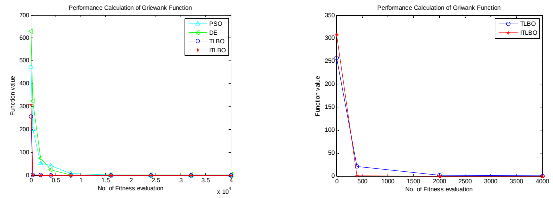
f_{14}



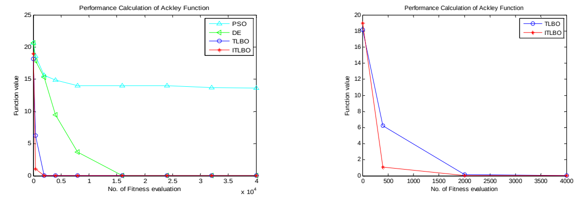
f15



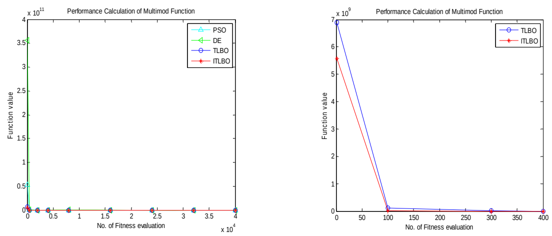
f16



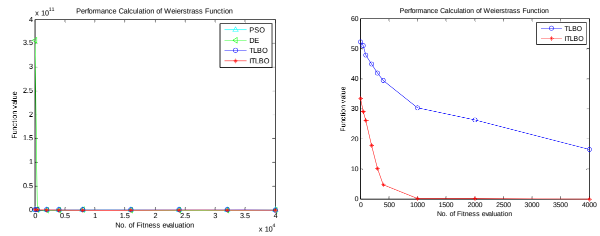
f17



f18



f19



f20

4.1.3 Results

In Experiments 1, we compared the PSO, DE, TLBO and ITLBO algorithms on a large set of functions described in the previous section and are listed in Tables 1. Each of the experiments in this section was repeated 30 times with different random seeds and they were terminated when they reached the maximum number of function evaluations or when they reached the global minimum value. Mean and standard deviation of fitness values on 30 different run produced by the algorithms have been recorded in the table 2 and at the same time mean value and standard deviation of no of fitness evaluation produced by the algorithms have been recorded in the Table 3.

In order to analyze the results whether there is significance between the results of each algorithm, we performed t-test on pairs of algorithms which is quite popular among researchers in evolutionary computing (Swagatam Das et al., 2009). In the table 4 we report the statistical significance level of difference of the means of PSO and ITLBO algorithm, DE and ITLBO algorithm, TLBO and ITLBO algorithm. Note that here '+' indicates the *t* value is significant at a 0.05 level of significance by two-tailed test, '·' means the difference of means is not statistically significant and 'NA' stands for Not Applicable, covering cases for which the two algorithms achieve the same accuracy results. From the Table 4, we understand that in 15 cases ITLBO performs better than PSO, where as in 14 cases ITLBO performs better than DE and in 8 cases ITLBO performs better than TLBO. To show convergence nature of each algorithm corresponding to each function we have made graph for each function. We have drawn two different graphs for each function. In the first graph we have shown convergence nature of all four algorithms where as in second graph we have shown the convergence nature of TLBO and ITLBO, as ITLBO is modification of TLBO to improve convergence characteristic. From Figs 1-20, it is clear that in almost all cases ITLBO have faster convergence than others.

4.2. Experiments 2: ITLBO vs. OEA, HPSO-TVAC, CLPSO and APSO

The experiments in this section constitute the comparison of the ITLBO algorithm versus OEA, HPSO-TVAC, CLPSO and APSO on 8 benchmarks function described in Table 1. The experimental result for OEA, HPSO-TVAC, CLPSO and APSO are gained from (Zhan et al., 2009; Ratnaweera et al., 2004) directly , where OEA uses the number of 3.0×10^5 FEs and HPSO-TVAC, CLPSO and APSO use the number of 2.0×10^5 FEs, where as ITLBO runs for 4.0×10^4 FEs. In the last column of table 5 shows the significance level between best and second best algorithm using t test at a 0.05 level of significance by two tailed test. Note that here '+' indicates the t value is significant ,'.' means the difference of means is not statistical significant and 'NA' stands for Not applicable, covering cases for which the two algorithms achieve the same accuracy results. As can be seen from table 5, ITLBO greatly outperforms OEA, HPSO-TVAC, CLPSO and APSO with better mean and standard deviation.

Table 5

Performance comparison for ITLBO,OEA,HPSO-TVAC,CLPSO and APSO

Function		OEA	HPSO-TVAC	CLPSO	APSO	ITLBO	Significant
Sphere	Mean	2.48e-30	3.38e-41	1.89e-19	1.45e-150	0	+
	Std	1.128e-29	8.50e-41	1.49e-19	5.73e-150	0	
Schwefel 2.22	Mean	2.068e-13	6.9e-23	1.01e-13	5.15e-84	1.0079e-238	+
	Std	2.440e-12	6.89e-23	6.54e-14	1.44e-83	0	
Schwefel 1.2	Mean	1.883e-09	2.89e-07	3.97e+02	1.0e-10	0	+
	Std	3.726e-9	2.97e-07	1.42e+02	2.13e-10	0	
Step	Mean	0	0	0	0	0	NA
	Std	0	0	0	0	0	
Rastrigin	Mean	5.430e-17	2.39	2.57e-11	5.8e-15	0	+
	Std	1.683e-16	3.71	6.64e-11	1.01e-14	0	
Noncontinous	Mean	N	1.83	0.167	4.14e-16	0	+
Rastrigin	Std	N	2.65	0.379	1.45e-15	0	
Ackley	Mean	5.336e-14	2.06e-10	2.01e-12	1.11e-14	1.7702e-15	NA
	Std	2.945e-13	9.45e-10	9.22e-13	3.55e-15	1.2434e-15	
Griewank	Mean	1.317e-02	1.07e-02	6.45e-13	1.67e-02	0	+
	Std	1.561e-02	1.14e-02	2.07e-12	2.41e-02	0	

4.3 Experiment 3: ITLBO vs. JADE, jDE and SaDE

The experiments in this section constitute the comparison of the ITLBO algorithm versus SaDE, jDE and JADE on 8 benchmark functions which are describe in Table 1. The results of JADE, jDE and SaDE are gained from (Z. H. Zhan et al., 2009) directly. In the last column of Table 6 shows the significance level between best and second best algorithm using t test at a 0.05 level of significance by two tailed test. Note that here '+' indicates the t value is significant,'.' means the difference of means is not statistical significant and 'NA' stands for Not applicable, covering cases for which the two algorithms achieve the same accuracy results. It can be seen from Table 6 that ITLBO performs much better than these DE variants on almost all the functions .

Table 6

Performance comparisons ITLBO, IADE, jDE and SaDE

Function	FEs		SaDE	jDE	JADE	ITLBO	Significant
Sphere	1.5×10^5	Mean	4.5e-20	2.5e-28	1.8e-60	0	+
		Std	1.9e-14	3.5e-28	8.4e-60	0	
Schwefel 2.22	2.0×10^5	Mean	1.9e-14	1.5e-23	1.8e-25	0	+
		Std	1.1e-14	1.0e-23	8.8e-25	0	
Schwefel 1.2	5.0×10^5	Mean	9.0e-37	5.2e-14	5.7e-61	0	+
		Std	5.4e-36	1.1e-13	2.7e-60	0	
Step	1.0×10^4	Mean	9.3e+02	1.0e+03	2.9e+00	0	+
		Std	1.8e+02	2.2e+02	1.2e+00	0	
Rastrigin	1.0×10^5	Mean	1.2e-03	1.5e-04	1.0e-04	0	+
		Std	6.5e-04	2.0e-04	6.0e-05	0	
Schwefel 2.21	5.0×10^5	Mean	7.4e-11	1.4e-15	8.2e-24	0	+
		Std	1.82e-10	1.0e-15	4.0e-23	0	
Ackley	5.0×10^4	Mean	2.7e-03	3.5e-04	8.2e-10	4.1892e-14	+
		Std	5.1e-04	1.0e-04	6.9e-10	2.8235e-14	
Griewank	5.0×10^4	Mean	7.8e-04	1.9e-05	9.9e-08	0	+
		Std	1.2e-03	5.8e-05	6.0e-07	0	

4.4 Experiment 4: ITLBO vs. CABC, GABC ,RABC and IABC

The experiments in this section constitute the comparison of the ITLBO algorithm versus CABC (Alatas et al., 2010), GABC (Zhu et al., 2010), RABC (Kang et al., 2011) and IABC (Weifeng Gao et al., 2011) on 8 benchmark functions which are described in table 1 . The parameters of the algorithms are identical to (Kang et al., 2011). In the last column of table 7 shows the significance level between best and second best algorithm using t test at a 0.05 level of significance by two tailed test. Note that here '+' indicates the t value is significant ,'.' means the difference of means is not statistical significant and 'NA' stands for Not applicable, covering cases for which the two algorithms achieve the same accuracy results The results, which have been summarized in Table 7, show that ITLBO performs much better in most cases than these ABC variants.

Table 7
Performance comparison of ITLBO, CABC, GABC, RABC and IABC

Function	Fes		CABC	GABC	RABC	IABC	ITLBO	Significant
Sphere	$1.5\text{x}10^5$	Mean	2.3e-40	3.6e-63	9.1e-61	5.34e-178	0	+
		Std	1.7e-40	5.7e-63	2.1e-60	0	0	
Schwefel 2.22	$2.0\text{x}10^5$	Mean	3.5e-30	4.8e-45	3.2e-74	8.82e-127	0	+
		Std	4.8e-30	1.4e-45	2.0e-73	3.49e-126	0	
Schwefel 1.2	$5.0\text{x}10^5$	Mean	8.4e+02	4.3e+02	2.9e-24	1.78e-65	0	+
		Std	9.1e+02	8.0e+02	1.5e-23	2.21e-65	0	
Step	$1.0\text{x}10^4$	Mean	0	0	0	0	0	NA
		Std	0	0	0	0	0	
Rastrigin	$5.0\text{x}10^4$	Mean	1.3e-00	1.5e-10	2.3e-02	0	0	+
		Std	2.7e-00	2.7e-10	5.1e-01	0	0	
Schwefel 2.21	$5.0\text{x}10^5$	Mean	6.1e-03	3.6e-06	2.8e-02	4.98e-38	0	+
		Std	5.7e-03	7.6e-07	1.7e-02	8.59e-38	0	
Ackley	$5.0\text{x}10^4$	Mean	1.0e-05	1.8e-09	9.6e-07	3.87e-14	2.4672e-15	+
		Std	2.4e-06	7.7e-10	8.3e-07	8.52e-15	1.8165e-15	
Griewank	$5.0\text{x}10^4$	Mean	1.2e-04	6.0e-13	8.7e-08	0	0	+
		Std	4.6e-04	7.7e-13	2.1e-08	0	0	

The t-test results also clearly indicate that the difference between the ITLBO algorithm and the other algorithms is statistically significant in most cases. Therefore, it is evident that the ITLBO algorithm has a good performance.

5. Conclusion

In this paper, we have developed a novel algorithm, ITLBO, to solve global numerical optimization problems by introducing a new search mechanism. We testify the performance of the proposed approach on a numbers of benchmark functions and provide comparisons with some other algorithms. The results show that the ITLBO algorithm possesses superior performance in accuracy, convergence, speed, stability and robustness, as compared to the other algorithms. Hence, the ITLBO algorithm may be a good alternative to deal with complex numerical optimization problems. Practical applications of the proposed approach in areas of clustering, data mining, design and optimization of communication networks, would also be worth studying.

References

Alatas, A. (2010). Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications*, 37, 5682–5687.

Das, S., Abraham, A., & Konar A. (2008). Automatic Clustering Using an Improved Differential Evolution Algorithm. *IEEE transactions on systems, man, and cybernetics*—part a: systems and humans, 38(1).

Das, S., Abraham, A., Chakraborty, U.K., & Konar, A. (2009). Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computations*, 13, 526–553.

Gao, W., & Liu, S. (2011). Improved artificial bee colony algorithm for global optimization. *Information Processing Letters,* 111, 871–882.

Krishnanand, K.R., Panigrahi, B. K., Rout, P.K., & Mohapatra, A. (2011). Application of multi-objective teaching learning based algorithm to an economic load dispatch problem with incommensurable objectives. *Lecture Notes in Computer Science,* SEMCCO 2011, Part I, 7076, 697-705.

Kang, F., Li, J. J., & Ma, Z.Y. (2011). Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. *Information Sciences*, 12, 3508–3531.

Naik, A., K., Parvathi, Satapathy, S. C., Nayak, R., & Panda, B. S. (2012). QoS multicast routing using teaching learning based optimization, ICAdC 2012. *Advances in Intelligent and Soft Computing Series*, Springer.

Naik, A., Satapathy, S. C. Rough set and Teaching learning based optimization technique for Optimal Features Selection. *Central European Journal of Computer Science*, to appear.

Naik, A., Satapathy, S. C., Parvathi, K. (2012). Improvement of initial cluster center of c- means using teaching learning based optimization. *Procedia Technology*, to appear.

Rao, R.V., Savsani, V.J., 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., & Vakharia, D.P. (2012). Teaching-learning-based optimization: a novel optimization method for continuous non-linear large scale problems. *Information Sciences*, 183, 1–15.

Rao, R.V., & Patel, V. (2012). An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations*, 3(4), 535-560.

Rao, R.V.,& Patel, V.K. (2012). Multi-objective optimization of combined Brayton and inverse Brayton cycles using advanced optimization algorithms. *Engineering Optimization*, 44(8), 965-983.

Rao, R.V., & Savsani, V.J. (2012). *Mechanical design optimization using advanced optimization techniques*. Springer-Verlag London, UK.

Rao, R. V., & Kalyankar, V. D. (2012). Parameter optimization of machining processes using a new optimization algorithm. *Materials and Manufacturing Processes*, 27(9), 978-985.

Ratnaweera, A., Halgamuge, S., Watson, H. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computers*, 8, 240–255

Satapathy, S. C., & Naik, A. (2011). Data clustering using teaching learning based Optimization. SEMCCO 2011. Part II, *Lecture Notes in Computer Science*, 7077, 148–156.

Satapathy, S.C., Naik, A., K., Parvathi (2012). High dimensional real parameter optimization with teaching learning based optimization. *International Journal of Industrial Engineering Computations*, 3(5), 807-816.

Satapathy, S.C., Naik., A., & Parvathi, K. (2012). 0-1 integer programming for generation maintenance scheduling in power systems based on teaching learning based optimization (TLBO). IC3 2012. *Communications in Computer and Information Science (CCIS) series*, to appear.

Toğan, V. (2012). Design of planar steel frames using Teaching–Learning Based Optimization. *Engineering Structures*. 34, 225–232.

Zhan, Z.H., Zhang, J., Li, Y., Chung, S.H. (2009). Adaptive particle swarm optimization. *IEEE Transactions in Systems Man Cybernetic B Cybernetic*, 39(6), 1362–1381.

Zhu, G.P., & Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function. *Applied Soft Computing*, 10(2), 445-456.