Research Article

An Improved Teaching-Learning-Based Optimization with Differential Learning and Its Application

Feng Zou,^{1,2} Lei Wang,¹ Debao Chen,² and Xinhong Hei¹

¹*School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China*

²*School of Physics and Electronic Information, Huaibei Normal University, Huaibei 235000, China*

Correspondence should be addressed to Lei Wang; leiwang@xaut.edu.cn

Received 22 April 2015; Revised 24 June 2015; Accepted 12 July 2015

Academic Editor: Ralph B. Dinwiddie

Copyright © 2015 Feng Zou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The teaching-learning-based optimization (TLBO) algorithm is a population-based optimization algorithm which is based on the effect of the influence of a teacher on the output of learners in a class. A variant of teaching-learning-based optimization (TLBO) algorithm with differential learning (DLTLBO) is proposed in the paper. In this method, DLTLBO utilizes a learning strategy based on neighborhood search of teacher phase in the standard TLBO to generate a new mutation vector, while utilizing a differential learning to generate another new mutation vector. Then DLTLBO employs the crossover operation to generate new solutions so as to increase the diversity of the population. By the integration of the local search and the global search, DLTLBO achieves a tradeoff between exploration and exploitation. To demonstrate the effectiveness of our approaches, 24 benchmark functions are used for simulating and testing. Moreover, DLTLBO is used for parameter estimation of digital IIR filter and experimental results show that DLTLBO is superior or comparable to other given algorithms for the employed examples.

1. Introduction

TLBO is a recently proposed population-based algorithm which simulates the teaching-learning process in a classroom. Rao et al. proposed TLBO for the optimization of mechanical design problems [1] and then applied TLBO to find global solutions to large-scale nonlinear optimization problems [2]. Toğan [3] presented a design procedure that employed TLBO for the discrete optimization of planar steel frames. In [4], elitist concept is introduced in the TLBO algorithm and its effect on the performance of the algorithm is investigated. The effects of common controlling parameters such as the population size and the number of generations on the performance of the algorithm are also investigated. Degertekin and Hayalioglu [5] applied TLBO in the optimization of four truss structures. Rao and Patel [6] proposed an improved TLBO by introducing the concepts of the number of teachers, adaptive teaching factor, tutorial training, and self-motivated learning. Niknam et al. [7] presented a θ -multiobjective TLBO algorithm to solve the dynamic economic emission dispatch problem, where the optimization procedure considers phase angles attributed to the real value of the design

parameters, instead of the design parameters themselves. Zou et al. [8] proposed an improved teaching-learning-based optimization algorithm with dynamic group strategy for global optimization problems. In [9], a ring neighborhood topology is introduced into the original TLBO algorithm to maintain the exploration ability of the population. Details of the conceptual basis of TLBO were given by Waghmare [10]. TLBO has emerged as one of the simplest and most efficient techniques, as it has been empirically shown to perform well on many optimization problems. However, in evolutionary computation research there have been always attempts to improve any given findings further and further.

The problem of IIR system identification can also be viewed as a problem of parameter estimation of adaptive digital filtering. For adaptive digital IIR filter design, the objective of the adaptation is to adjust filter coefficients of a digital filter to estimate actual parameter values of an unknown system from its inputs and outputs. However, there are some potential disadvantages with the design of adaptive digital IIR filters [11–13]. A major concern in these disadvantages is the mean square error between the desired response and estimated filter output of digital IIR filter. This error surface

is generally nonquadratic and multimodal with respect to the filter coefficients, and hence optimization algorithms are required to find out better global solution. In order to avoid the local optima problem encountered by gradient descent methods in IIR system identification, considering IIR system identification as an optimization problem, recently, evolutionary algorithm and swarm intelligence, such as Genetic Algorithms (GA) [14–16], Simulated Annealing (SA) [17], Tabu Search (TS) [18], Differential Evolution (DE) [19–21], Ant Colony Optimization (ACO) [22], Artificial Bee Colony (ABC) algorithm [23], Particle Swarm Optimizer (PSO) [24, 25], Gravitation Search Algorithm (GSA) [26, 27], and Cuckoo search optimization [28], have been made for studying alternative structures and algorithms for adaptive digital IIR filters.

In order to find suitable filter coefficients for the digital IIR filter efficiently, this paper aims to develop a teaching-learning-based optimization (TLBO) approach. First, an improved TLBO with differential learning (DLTLBO) is proposed in paper. In the method, DLTLBO utilizes the integration of the learning strategy based on neighborhood search of teacher phase in the standard TLBO and the differential learning to generate two new mutation vectors, and DLTLBO employs the crossover operation to generate new solutions so as to increase the diversity of the population. Then, to evaluate the optimization performance of the proposed algorithm, the comparison study on 24 benchmark optimization problems is given. Moreover, DLTLBO is applied to the design of the digital IIR filter, and its performance is compared to that of other algorithms. The simulation results show that DLTLBO has better or comparable performance compared to the other algorithms for the employed examples.

The remainder of this paper is organized as follows. Section 2 describes the teaching-learning-based optimization. Section 3 presents the improved teaching-learning-based optimization with differential learning (DLTLBO). In Section 4, 24 benchmark functions are tested and the experiments are conducted along with statistical tests. DLTLBO is applied to the design of the digital IIR filter in Section 5. Conclusions are given in Section 6.

2. Teaching-Learning-Based Optimization

Inspired by the philosophy of the classical school teaching and learning process, Rao et al. [1, 2] first proposed a novel teaching-learning-based optimization (TLBO). In this method, population consists of learners in a class and design variables are courses offered. The brief description of the algorithm is given as follows.

2.1. Teacher Phase. During the teacher phase, learner will move their position toward the position of the best learner (*Teacher*), by taking into account the current mean value of the learners (*Mean*) that represents the average qualities of all learners in the population. The updating formula of the learning for a learner X_i in teaching phase is given by

$$newX_i = X_i + r \times (Teacher - TF \times Mean), \quad (1)$$

where TF is a teaching factor that decides the value of *Mean* to be changed and r is a random vector where each element is a random number in the range $[0, 1]$. The value of TF can be either 1 or 2, which is again a heuristic step and is decided randomly with equal probability as

$$TF = \text{round} [1 + \text{rand} (0, 1)] . \quad (2)$$

Learner modification is expressed as follows:

for each learner X_i

$$TF = \text{round}[1 + \text{rand}(0, 1)]$$

$$newX_i = X_i + r \times (Teacher - TF \times Mean)$$

Accept $newX_i$ if it gives a better function value

endfor

2.2. Learner Phase. During the learner phase, learners increase their knowledge by interacting with each other. A learner X_i learns something new if the other X_j learner has more knowledge than him or her:

$$newX_i = \begin{cases} X_i + r * (X_i - X_j) & \text{if } f(X_i) < f(X_j) \\ X_i + r * (X_j - X_i) & \text{otherwise,} \end{cases} \quad (3)$$

where r is a random vector where each element is a random number in the range $[0, 1]$, $f(X_i)$ is the fitness value of the learner X_i , and $f(X_j)$ is the fitness value of the learner X_j .

Learner modification is expressed as follows:

for each learner X_i of the class

Randomly select one learner X_k , such that $k \neq i$

$$newX_i = \{X_i + r * (X_i - X_j), \text{ if } f(X_i) < f(X_j); \\ X_i + r * (X_j - X_i), \text{ otherwise}\}$$

Accept $newX$ if it gives a better function value

end for

3. Proposed Algorithm: DLTLBO

The search in TLBO is mainly based on the global information, and DE is mainly based on the distance and direction information which is a kind of local information [29]. To balance the global and local search ability, a modified interactive learning strategy is proposed in teacher phase. In this method, DLTLBO utilizes the learning strategy based on neighborhood search of teacher phase in the standard TLBO to generate a new mutation vector, while utilizing the differential learning to generate another new mutation vector. Different from our previous method [30], DLTLBO employs the DE crossover operation to generate new solutions so as to increase the diversity of the population. The complete flowchart of the DLTLBO algorithm is shown in Figure 1.

In the proposed DLTLBO algorithm, for each learner X_i , its associated mutant vector U_i based on differential mutation can be generated as follows:

$$U_i = X_i + r \times (NTeacher - TF \times NMean), \quad (4)$$

FIGURE 1: Flowchart showing the working of DTLBO algorithm.

where $NTeacher$ and $NMean$ are the teacher and the mean of the learner's corresponding neighborhood, respectively. In this paper, ring neighborhood topology with k -neighborhood radius is used to determine in the neighborhood area of each learner.

For each learner X_i , its other associated mutant vector V_i can be generated based on differential mutation as follows:

$$V_i = X_{r1} + F * (X_{r2} - X_{r3}). \quad (5)$$

After the mutation phase, crossover operation is applied to each pair of the mutant vectors in order to enhance the potential diversity of the population. That is, the updating

formula of the learning for a learner X_i in teacher phase is given as follows:

$$newX_{ij} = \begin{cases} V_{ij} & \text{if } \text{rand}[0, 1] \leq CR \\ U_{ij} & \text{otherwise.} \end{cases} \quad (6)$$

CR is a parameter called crossover probability; U_{ij} is the learners which is calculated according to (4) and V_{ij} is the learners which is calculated according to (5).

Finally, the one-to-one greedy selection is employed by means of comparing a parent X_i and its corresponding offspring $newX_i$.

In learner phase, interactive learning of the original TLBO algorithm in the learner phase is still used in our DTLBO algorithm. In this learning method, one of the

```

(1) Begin
(2) Initialize  $N$  (number of learners),  $D$  (number of dimensions)  $F$  and CR
(3) Initialize learners  $X$  and evaluate all learners  $X$ 
(4) Donate  $NTeacher$  and  $NMean$  of the learner's corresponding neighborhood
(5) while (stopping condition not met)
(6)   for each learner  $X_i$  of the class           % Teacher phase
(7)      $TF = \text{round}(1 + \text{rand}(0, 1))$ 
(8)     for  $j = 1 : D$ 
(9)        $U_{ij} = X_{ij} + \text{rand} * (NTeacher_j - TF * NMean_j)$ 
(10)    endfor
(11)    Select three learners  $X_{r1}$   $X_{r2}$   $X_{r3}$  randomly from the current class where  $i \neq r1 \neq r2 \neq r3$ 
(12)     $V_i = X_{r1} + F * (X_{r2} - X_{r3})$ 
(13)    for  $j = 1 : D$ 
(14)      if  $\text{rand}(0, 1) < CR$ ,  $newX_{ij} = V_{ij}$ ; else,  $newX_{ij} = U_{ij}$ ; endif
(15)    endfor
(16)    Accept  $newX_i$  if  $f(newX_i)$  is better than  $f(X_i)$ 
(17)  endfor
(18)  for each learner  $X_i$  of the class           % Learner phase
(19)    Randomly select one learner  $X_k$ , such that  $i \neq k$ 
(20)    if  $f(X_i)$  better  $f(X_k)$ 
(21)      for  $j = 1 : D$ 
(22)         $newX_{ij} = X_{ij} + \text{rand} * (X_{ij} - X_{kj})$ 
(23)      endfor
(24)    else
(25)      for  $j = 1 : D$ 
(26)         $newX_{ij} = X_{ij} + \text{rand} * (X_{kj} - X_{ij})$ 
(27)      endfor
(28)    endif
(29)    Accept  $newX_i$  if  $f(newX_i)$  is better than  $f(X_i)$ 
(30)  endfor
(31) endwhile
(32) end

```

ALGORITHM 1: DTLBO(-).

learners randomly learns from the other learner in the population. This learning method can be treated as the global search strategy. As explained above, the pseudocode for the implementation of DTLBO is summarized in Algorithm 1.

As explained in Algorithm 1, it can be observed that the probability CR determines adopting the learning strategy of the learners in the teacher phase of the DTLBO algorithm. That is, a random number r between 0 and 1 is generated for each learner; if $r < CR$, the learning strategy of the original TLBO will be adopted by the learner; otherwise, the differential learning strategy will be adopted by the learner. This will be helpful in balancing the advantages of fast convergence rate (the attraction of the learning strategy of TLBO learning) and exploration for local areas (neighborhood search strategy and the differential learning) in DTLBO. By this means, DTLBO achieves a tradeoff between exploration and exploitation, improves the global search capability, and accelerates the convergence.

4. Functions Optimization

In this section, to compare the performance of DTLBO with some other methods, 7 algorithms are also simulated in the

paper. The details of benchmark functions, the simulation settings, and experimental results are given as follows.

4.1. Benchmark Functions. The details of 24 benchmark functions are shown in Table 1. Among 24 benchmark functions, F_1 to F_4 are unimodal functions, F_5 to F_{10} are multimodal functions, and F_{11} to F_{18} are the rotated version of F_3 to F_{10} , respectively. F_{19} to F_{24} are taken from the CEC 2008 test suite. The searching range and theory optima for all functions are also shown in Table 1.

4.2. Parameter Settings. In this paper, all the experiments are carried out on the same machine with a Celoron 2.26 GHz CPU, 2 GB memory, and Windows XP operating system with MATLAB 7.9. For the purpose of reducing statistical errors, each function is independently simulated 50 runs, and their mean results are used in the comparison. For all approaches, the population size was set to 50, and the maximal number of cost function evaluations (FEs) is used as termination condition of algorithm, namely, 100,000 for 10D problems. The parameter settings for all algorithms in comparison are extracted from their corresponding literatures and are described as follows:

TABLE 1: Details of numerical benchmarks used.

Function	Formula	Range	Optima
Sphere	$F_1(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]$	0
Quadric	$F_2(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]$	0
Sum square	$F_3(x) = \sum_{i=1}^D i x_i^2$	$[-10, 10]$	0
Zakharov	$F_4(x) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5 i x_i \right)^2 + \left(\sum_{i=1}^D 0.5 i x_i \right)^4$	$[-10, 10]$	0
Rosenbrock	$F_5(x) = \sum_{i=1}^{D-1} \left[100 (x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right]$	$[-2.048, 2.048]$	0
Ackley	$F_6(x) = 20 - 20 \exp \left(-\frac{1}{5} \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + e$	$[-32.768, 32.768]$	0
Rastrigin	$F_7(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0
Weierstrass	$F_8(x) = \sum_{i=1}^D \left(\sum_{k=0}^{kmax} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k \times 0.5)]$ $a = 0.5 \quad b = 3 \quad kmax = 20$	$[-0.5, 0.5]$	0
Griewank	$F_9(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]$	0
Schwefel	$F_{10}(x) = 418.9829D + \sum_{i=1}^D \left(-x_i \sin \sqrt{abs(x_i)} \right)$	$[-500, 500]$	0
Rotated sum square	$F_{11}(x) = \sum_{i=1}^D i y_i^2 \quad y = M \times x$	$[-10, 10]$	0
Rotated Zakharov	$F_{12}(x) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5 i y_i \right)^2 + \left(\sum_{i=1}^D 0.5 i y_i \right)^4 \quad y = M \times x$	$[-10, 10]$	0
Rotated Rosenbrock	$F_{13}(x) = \sum_{i=1}^{D-1} \left[100 (y_i^2 - y_{i+1})^2 + (y_i - 1)^2 \right] \quad y = M \times x$	$[-2.048, 2.048]$	0
Rotated Ackley	$F_{14}(x) = 20 - 20 \exp \left(-\frac{1}{5} \sqrt{\frac{1}{D} \sum_{i=1}^D y_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi y_i) \right) + e \quad y = M \times x$	$[-32.768, 32.768]$	0
Rotated Rastrigin	$F_{15}(x) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10) \quad y = M \times x$	$[-5.12, 5.12]$	0
Rotated Weierstrass	$F_{16}(x) = \sum_{i=1}^D \left(\sum_{k=0}^{kmax} [a^k \cos(2\pi b^k (y_i + 0.5))] \right) - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k \times 0.5)]$ $a = 0.5 \quad b = 3 \quad kmax = 20 \quad y = M \times x$	$[-0.5, 0.5]$	0
Rotated Griewank	$F_{17}(x) = \sum_{i=1}^D \frac{y_i^2}{4000} - \prod_{i=1}^n \cos \left(\frac{y_i}{\sqrt{i}} \right) + 1 \quad y = M \times x$	$[-600, 600]$	0
Rotated Schwefel	$F_{18}(x) = 418.9829D + \sum_{i=1}^D \left(-y_i \sin \sqrt{abs(y_i)} \right) \quad y = M \times x$	$[-500, 500]$	0
Shifted sphere	$F_{19}(x) = \sum_{i=1}^D z_i^2 - 450$ $z = x - o \quad o: \text{the shifted global optimum}$	$[-100, 100]$	o
Shifted Schwefel's Problem 2.21	$F_{20}(x) = \max_i \{ z_i , 1 \leq i \leq D\} - 450$ $z = x - o + 1 \quad o: \text{the shifted global optimum}$	$[-100, 100]$	o
Shifted Rosenbrock	$F_{21}(x) = \sum_{i=1}^{D-1} \left[100 (z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right] + 390$ $z = x - o + 1 \quad o: \text{the shifted global optimum}$	$[-100, 100]$	o
Shifted Rastrigin	$F_{22}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) - 330$ $z = x - o \quad o: \text{the shifted global optimum}$	$[-5, 5]$	o

TABLE 1: Continued.

Function	Formula	Range	Optima
Shifted Griewank	$F_{23}(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 - 180$ $z = x - o \quad o: \text{the shifted global optimum}$	$[-600, 600]$	o
Shifted Ackley	$F_{24}(x) = 20 - 20 \exp\left(-\frac{1}{5} \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + e - 140$ $z = x - o \quad o: \text{the shifted global optimum}$	$[-32, 32]$	o

- (i) jDE [31]: $F = 0.5$, $CR = 0.9$;
- (ii) SaDE [32]: $F \sim N(0.5, 0.3)$, $CR_0 = 0.5$, $CR \sim N(CR_0, 0.1)$, $LP = 50$;
- (iii) PSO-*w*FIPS [33]: $w = 0.7298$;
- (iv) PSO-FDR [34]: $w_{min} = 0.4$, $w_{max} = 0.9$, $\psi_1 = 1$, $\psi_2 = 1$, $\psi_3 = 2$;
- (v) ETLBO [4]: elite size = 2;
- (vi) Proposed DTLBO: $F = 0.5$, $CR = 0.9$, neighborhood size = 3.

4.3. Comparisons on the Solution Accuracy. The results are shown in Table 2 in terms of the average best solution and the standard deviation of the solutions obtained in the 50 independent runs by each algorithm for 10D problems on 24 test functions, where “ $w/t/l$ ” summarizes the competition results among DTLBO and other algorithms in the last row of the table, which means that DTLBO wins in w functions, ties in t functions, and loses in l functions. The best results among the algorithms are shown in bold. Figure 2 presents the convergence graphs of different benchmark functions in terms of the median fitness values achieved by 7 algorithms for 50 runs.

The comparisons in Table 2 show that DTLBO outperforms jDE on 10 functions, while jDE achieves better results on 10 functions. SaDE obtains better performance on 11 functions, while DTLBO only outperforms jDE for 9 functions. PSO-*w*FIPS achieves better results than DTLBO on 5 functions, while DTLBO performs better for the remaining 19 functions. FDR-PSO only obtains better performance than DTLBO on F_5 , F_{20} , and F_{21} , while DTLBO outperforms FDR-PSO for the remaining 19 functions except F_6 and F_8 . DTLBO outperforms TLBO on 16 functions and outperforms ETLBO on 17 functions, while TLBO, ETLBO, and DTLBO perform well on functions F_8 and F_{16} . It can be concluded that the proposed DTLBO significantly improves the algorithm’s performance, although DTLBO is not always better than other algorithms.

5. IIR System Identification Using DTLBO

The goal of filter modeling is to alter filter coefficients of a digital filter to match an unknown system transfer function. Evolutionary algorithms are heuristic search methods, which are characterized by features as stochastic, adaptive, and learning in order to produce intelligent optimization

schemes. In this section, DTLBO is used to optimize the coefficients vector of the digital IIR filter.

5.1. Digital IIR Filter Identification. Application of IIR filter in system identification has been widely studied since many problems encountered in signal processing can be characterized as a system identification problem. Therefore, in the simulation study, digital IIR filters are designed for the system identification purpose. In this case, the parameters of the filters are successively adjusted by evolutionary optimization algorithms until the error between the output of the filter and the unknown system is minimized. In general, we expect that the number of zero points is more than the number of pole points. A schematic of filter modeling problem using the evolutionary algorithms is shown in Figure 3.

The general expression of digital IIR filters is given by the following difference equation [35]:

$$y(k) = \sum_{i=0}^M a_i x(k-i) - \sum_{i=1}^N b_i y(k-i), \quad (7)$$

where $x(k)$ is the input of the digital IIR filter at time k , $y(k)$ is the output of the digital IIR filter at time k , M and N are denominator and numerator orders, N ($N \geq M$) is the filter’s order, a_i and b_i are adjustable filter coefficients of the model, and $b_0 = 1$. The number of the digital IIR filter coefficients is $(N + M + 1)$.

For adaptive digital IIR filter design, the objective of the adaptation is to adjust filter coefficients of a digital filter to estimate actual parameter values of an unknown system from its inputs and outputs. The design of this filter can be considered as a minimization problem of the cost function. The most commonly used approach to digital IIR filter design is to formulate the problem as an optimization problem with the mean square error (MSE) as the cost function to be minimized:

$$J(v) = E[e^2(k)]$$

$$= \frac{1}{S} \sum_{k=1}^S [(idealH(k) - actualH(k))^2], \quad (8)$$

where $v = [a_0 \ a_1 \ \dots \ a_M \ b_1 \ b_2 \ \dots \ b_N]$ denotes the digital IIR filter coefficient vector, S is the number of samples used for the computation of the time-averaged cost function, and $ideal(k)$ and $actual(k)$ are the filter’s ideal and actual responses, respectively.

TABLE 2: Mean $\pm$ Std. of the solutions for 10D problems.

Fun	jDE	SADE	PSO-wFIPS	FDR-PSO	TLBO	ETLBO	DLTLBO
F_1	$4.30e-093 \pm 4.93e-093$	$1.76e-088 \pm 3.90e-088$	$4.25e-020 \pm 2.62e-020$	$1.80e-110 \pm 2.36e-110$	$1.16e-220 \pm 0.00e+000$	$3.79e-196 \pm 0.00e+000$	$1.05e-262 \pm 0.00e+000$
F_2	$8.21e-036 \pm 1.59e-035$	$9.29e-044 \pm 2.07e-043$	$5.65e-008 \pm 2.66e-008$	$2.14e-044 \pm 4.75e-044$	$4.94e-097 \pm 7.18e-097$	$2.87e-094 \pm 3.82e-094$	$7.57e-119 \pm 9.09e-119$
F_3	$1.91e-095 \pm 3.85e-095$	$1.03e-090 \pm 2.19e-090$	$1.13e-021 \pm 1.09e-021$	$6.01e-112 \pm 8.88e-112$	$4.78e-221 \pm 0.00e+000$	$8.20e-198 \pm 0.00e+000$	$4.98e-265 \pm 0.00e+000$
F_4	$8.29e-039 \pm 1.83e-038$	$2.94e-059 \pm 3.98e-059$	$3.53e-013 \pm 1.94e-013$	$1.26e-061 \pm 1.87e-061$	$1.73e-104 \pm 3.80e-104$	$1.93e-100 \pm 2.26e-100$	$8.41e-129 \pm 2.64e-130$
F_5	$5.98e-019 \pm 1.24e-018$	$1.10e+000 \pm 1.54e+000$	$3.84e+000 \pm 1.57e-001$	$3.39e-002 \pm 1.19e-002$	$5.00e-003 \pm 1.01e-002$	$1.53e-003 \pm 2.30e-003$	$2.88e+000 \pm 4.11e-001$
F_6	$2.84e-015 \pm 1.59e-015$	$2.84e-015 \pm 1.59e-015$	$4.77e-011 \pm 1.47e-011$	$3.55e-015 \pm 0.00e+000$	$2.84e-015 \pm 1.59e-015$	$2.84e-015 \pm 1.59e-015$	$3.55e-015 \pm 0.00e+000$
F_7	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$8.30e-001 \pm 8.07e-001$	$1.99e+000 \pm 9.95e-001$	$5.88e-001 \pm 5.49e-001$	$2.19e+000 \pm 1.30e+000$	$0.00e+000 \pm 0.00e+000$
F_8	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$2.07e-010 \pm 1.90e-010$	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$
F_9	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$2.62e-002 \pm 1.29e-002$	$6.93e-002 \pm 4.92e-002$	$1.48e-003 \pm 3.31e-003$	$4.44e-017 \pm 9.93e-017$	$0.00e+000 \pm 0.00e+000$
F_{10}	$1.27e-004 \pm 0.00e+000$	$1.27e-004 \pm 0.00e+000$	$2.41e+002 \pm 7.65e+001$	$5.65e+002 \pm 1.88e+002$	$4.82e+002 \pm 2.04e+002$	$4.80e+002 \pm 1.81e+002$	$5.92e+001 \pm 8.37e+001$
F_{11}	$1.55e-070 \pm 3.46e-070$	$4.39e-066 \pm 9.81e-066$	$1.30e-019 \pm 1.51e-019$	$2.85e-102 \pm 6.37e-102$	$2.92e-209 \pm 0.00e+000$	$8.70e-190 \pm 0.00e+000$	$3.00e-250 \pm 0.00e+000$
F_{12}	$1.75e-041 \pm 1.99e-041$	$1.18e-053 \pm 2.64e-053$	$5.02e-013 \pm 3.20e-013$	$5.22e-061 \pm 1.15e-060$	$5.76e-106 \pm 1.24e-105$	$2.04e-100 \pm 4.28e-100$	$5.21e-128 \pm 7.36e-128$
F_{13}	$8.96e-011 \pm 2.00e-010$	$2.31e+000 \pm 2.19e+000$	$4.17e+000 \pm 2.89e-001$	$4.54e+000 \pm 3.59e+000$	$7.53e-002 \pm 8.83e-002$	$8.29e-001 \pm 1.78e+000$	$2.94e+000 \pm 1.62e+000$
F_{14}	$3.55e-015 \pm 0.00e+000$	$2.84e-015 \pm 1.59e-015$	$5.76e-011 \pm 2.92e-011$	$3.55e-015 \pm 0.00e+000$	$3.55e-015 \pm 0.00e+000$	$2.84e-015 \pm 1.59e-015$	$1.78e-015 \pm 2.51e-015$
F_{15}	$3.87e+000 \pm 1.28e+000$	$3.66e+000 \pm 1.55e+000$	$3.97e+000 \pm 3.24e+000$	$7.56e+000 \pm 3.76e+000$	$2.42e+000 \pm 1.81e+000$	$2.98e+000 \pm 1.41e+000$	$1.40e-004 \pm 1.97e-004$
F_{16}	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$1.48e-007 \pm 2.50e-007$	$7.13e-002 \pm 1.26e-001$	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$
F_{17}	$1.48e-003 \pm 3.31e-003$	$3.94e-003 \pm 5.67e-003$	$1.04e-001 \pm 8.74e-002$	$1.13e-001 \pm 5.80e-002$	$5.41e-003 \pm 8.59e-003$	$1.48e-003 \pm 3.31e-003$	$0.00e+000 \pm 0.00e+000$
F_{18}	$1.38e+002 \pm 1.22e+002$	$1.07e+002 \pm 1.21e+002$	$2.80e+002 \pm 1.28e+002$	$5.29e+002 \pm 2.00e+002$	$5.46e+002 \pm 3.23e+002$	$5.44e+002 \pm 2.81e+002$	$1.18e+002 \pm 0.00e+000$
F_{19}	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$2.21e-020 \pm 2.43e-020$	$3.03e+000 \pm 4.14e+000$	$0.00e+000 \pm 0.00e+000$	$1.01e-029 \pm 2.26e-029$	$2.52e-029 \pm 3.57e-029$
F_{20}	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$5.87e-008 \pm 2.46e-008$	$0.00e+000 \pm 0.00e+000$	$3.06e-011 \pm 5.16e-011$	$2.47e-010 \pm 5.33e-010$	$8.77e-001 \pm 1.23e+000$
F_{21}	$1.83e-017 \pm 3.99e-017$	$1.50e-001 \pm 2.13e-001$	$4.50e+000 \pm 1.57e-001$	$1.53e+000 \pm 2.06e+000$	$1.05e-001 \pm 1.49e-001$	$8.73e+000 \pm 1.78e+001$	$4.55e+000 \pm 8.73e-001$
F_{22}	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$2.26e+000 \pm 9.23e-001$	$3.58e+000 \pm 2.06e+000$	$7.76e+000 \pm 3.02e+000$	$9.55e+000 \pm 4.37e+000$	$3.48e+000 \pm 7.04e-001$
F_{23}	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$2.32e-002 \pm 1.71e-002$	$6.88e-001 \pm 5.35e-001$	$5.86e-002 \pm 2.77e-002$	$6.75e-002 \pm 3.32e-002$	$5.04e-002 \pm 3.65e-002$
F_{24}	$0.00e+000 \pm 0.00e+000$	$0.00e+000 \pm 0.00e+000$	$5.73e-011 \pm 2.17e-011$	$3.54e+000 \pm 1.98e+000$	$6.34e-001 \pm 9.19e-001$	$1.22e+000 \pm 7.14e-001$	$5.78e-001 \pm 8.17e-001$
w/t//	10/4/10	9/4/11	19/0/5	19/2/3	16/2/6	17/2/5

FIGURE 2: Continued.

FIGURE 2: Comparison of the performance curves of 7 algorithms.

To achieve higher stop band attenuation and to have better control on the transition width, the frequency cost function [27] to be minimized is defined as

$$J(v) = \frac{1}{S_p} \sum_{\omega_p} \text{abs} \left[\text{abs}(|\text{actual}H(\omega)| - 1) - \delta_p \right] + \frac{1}{S_s} \sum_{\omega_s} [\text{abs}(|\text{actual}H(\omega)| - \delta_s)], \quad (9)$$

where $\text{ideal}H(\omega)$ and $\text{actual}H(\omega)$ are the filter's ideal and actual frequency responses, respectively. δ_p and δ_s are the ripples in the pass band and stop band, ω_p and ω_s are pass band and stop band normalized cutoff frequencies,

respectively, and S_p and S_s are the number of samples used in the pass band and stop band. In this case, the best fitness $J(v) = 0$ represents that the magnitude response of the solution is within $[1 - \delta_p, 1]$ in the pass band and $[0, \delta_s]$ in the stop band.

5.2. Description of the DLTlBO for the Digital IIR Filter. The steps of the DLTlBO for the digital IIR filter are given as follows.

Step 1. Initialize every real coded learner $v = [a_0 \ a_1 \ \dots \ a_M \ b_1 \ b_2 \ \dots \ b_N]$ ($N \geq M$) of the population. Each learner consists of M numerator filter coefficients and N denominator filter coefficients; the total number of

TABLE 3: Design parameters of IIR LP, HP, BP, and BS filters.

Type of filter	Pass band ripple δ_p	Stop band ripple δ_s	Pass band edge ω_p	Stop band edge ω_s
Low-pass (LP)	0.01	0.001	0.45	0.5
High-pass (HP)	0.01	0.001	0.5	0.45
Band-pass (BP)	0.01	0.001	0.3 and 0.7	0.25 and 0.75
Band-stop (BS)	0.01	0.001	0.25 and 0.75	0.3 and 0.7

FIGURE 3: Filter design using a heuristic search algorithm.

filter coefficients is $N + M + 1$ for N th order filter to be designed. To maintain the stability of IIR filter, the efficient way of [17] is adopted. Hence, the coefficients a_i are formed by the constraints $[-2, 2]$ and all the lattice-reflection-form coefficients k_i of the direct-form coefficients b_i are less than 1.

Step 2. Compute the error fitness value for each learner according to the cost function.

Step 3 (teacher phase). Each learner learns from the teacher according to (4)–(6). Compute the error fitness value for each learner and update each learner.

Step 4 (learner phase). Each learner randomly learns from the other learner according to (3). Compute the error fitness value for each learner and update each learner.

Step 5 (termination criterion). Stop if the termination criterion is met; the dimensional values of the global teacher are the coefficients of the designed digital IIR filter; otherwise, repeat from Step 3.

5.3. Experimental Simulation. This section presents experimental evaluation of DTLBO on designing digital IIR filters. Some benchmark problems are selected to compare the performance of GA [36], PSOWFIPS [33], SaDE [32], TLBO [1, 2], and DTLBO algorithms. In all cases, the maximal value of FEs is used as ended condition of algorithm, namely, 25000 for GA, PSOWFIPS, SaDE, TLBO, and DTLBO, respectively; the number of population is set to 50, and the initial population is selected randomly. For GA, the crossover rate P_c is set to 0.9 and the mutation rate P_m is set to 0.1. For PSOWFIPS, $w = 0.7298$. For SaDE, $F \sim N(0.5, 0.3)$, $CR_0 = 0.5$, $CR \sim N(CR_m, 0.1)$, and $LP = 50$. For our DTLBO algorithm,

the scaling factor F is set to 0.9 and the crossover probability CR is set to 0.1.

Example 1. In the first example, the transfer function of the unknown plant is unknown and only the performance parameters are given in Table 3, including low-pass (LP), high-pass (HP), band-pass (BP), and band-stop (BS) filters.

In this experiment, the filter orders are set to 8 for all IIR LP, HP, BP, and BS filters and the number of samples is set to 128. The minimum and maximum values of the coefficients are -2 and 2 . The cost function is given as shown in (9). All the results are averaged over 20 random runs with randomly chosen initial positions. Table 4 shows the maximum values and the mean values of stop band attenuation, the means and standard deviations of MSE, and the means and standard deviations of CPU times. Figure 4 shows the comparison of convergence curves, magnitude responses, and gain responses of five algorithms. In addition, a statistical t -test [21] is given in Table 4.

As seen from Table 4, it is observed that the proposed 8th order IIR filter design using DTLBO spends the less CPU running time among all the three algorithms except GA. For low-pass (LP) filter, it is observed that the maximum stop band attenuations 28.13 dB, 30.22 dB, 31.43 dB, 28.74 dB, and 26.21 dB are obtained for GA, PSOWFIPS, SaDE, TLBO, and DTLBO, respectively. Statistical t -test between other algorithms and DTLBO also indicates that DTLBO performs well. Comparative curves in Figures 4(a)–4(c) also explore that the proposed 8th order IIR filter design using SaDE attains the highest stop band attenuation, although DTLBO obtains the least MSE. Figure 4(a) shows the convergence curves for the designed 8th order IIR LP filter obtained for different algorithms, Figure 4(b) shows the comparative magnitude plots in dB, and Figure 4(c) represents the comparative normalized gain plots for 8th order IIR LP filter. For high-pass (HP) filter and band-stop (BS) filter, it is seen that the proposed 8th order IIR filter design using DTLBO attains the highest stop band attenuation and the least MSE. It is also observed from Table 4 and Figure 4 that, for band-pass (BP) filter, the proposed 8th order IIR filter design using SaDE attains the highest stop band attenuation and the least MSE, and DTLBO obtains the second highest stop band attenuation and the second least MSE. It can be concluded that DTLBO has a good performance of the solution accuracy, although DTLBO needs more CPU running time.

TABLE 4: Statistical data for stop band attenuation, MSE, and CPU times for 8th order IIR filters.

Type of filter	Algorithm	Maximum gain (dB)	Mean gain (dB)	MSE	t-stat	p value	t-value	Time (s)
Low-pass (LP)	GA	28.13	40.37	$2.01e-001 \pm 3.40e-002$	13.60792	$3.01E-11$	2.093024	$4.24e+000 \pm 1.01e-002$
	PSOwFIPS	30.22	43.51	$1.10e-001 \pm 1.94e-002$	5.14552	$5.75E-05$	2.093024	$5.17e+000 \pm 4.69e-003$
	SaDE	31.43	48.23	$8.25e-002 \pm 3.25e-002$	1.31278	$2.05E-01$	2.093024	$5.77e+000 \pm 5.46e-002$
	TLBO	28.74	41.18	$9.40e-002 \pm 3.62e-002$	3.03097	$6.87E-03$	2.093024	$4.80e+000 \pm 2.12e-002$
High-pass (HP)	DLTLBO	26.21	38.37	$6.68e-002 \pm 3.13e-002$	—	—	—	$4.74e+000 \pm 1.17e-002$
	GA	16.85	28.28	$1.92e-001 \pm 4.54e-002$	12.53964	$1.23E-10$	2.09302	$4.26e+000 \pm 7.07e-003$
	PSOwFIPS	20.16	32.46	$1.24e-001 \pm 2.59e-002$	8.96138	$2.99E-08$	2.09302	$5.20e+000 \pm 6.41e-003$
	SaDE	28.49	38.72	$9.39e-002 \pm 4.11e-002$	2.64240	$1.61E-02$	2.09302	$5.70e+000 \pm 2.64e-002$
Band-pass (BP)	TLBO	25.39	35.37	$1.12e-001 \pm 2.89e-002$	5.73652	$1.58E-05$	2.09302	$4.78e+000 \pm 3.74e-002$
	DLTLBO	35.28	43.51	$6.12e-002 \pm 2.37e-002$	—	—	—	$4.73e+000 \pm 4.72e-003$
	GA	24.60	37.28	$3.24e-001 \pm 5.96e-002$	712807	$8.90E-07$	2.09302	$4.81e+000 \pm 1.21e-002$
	PSOwFIPS	24.23	36.84	$2.38e-001 \pm 2.91e-002$	3.50304	$2.38E-03$	2.09302	$5.72e+000 \pm 3.65e-002$
Band-stop (BS)	SaDE	29.73	42.19	$2.00e-001 \pm 2.77e-002$	-0.19447	0.84787	2.09302	$6.32e+000 \pm 4.39e-002$
	TLBO	25.91	39.52	$2.34e-001 \pm 4.00e-002$	2.80619	0.01127	2.09302	$5.37e+000 \pm 3.20e-002$
	DLTLBO	27.90	40.27	$2.01e-001 \pm 3.25e-002$	—	—	—	$5.28e+000 \pm 5.08e-003$
	GA	16.24	28.31	$3.62e-001 \pm 7.40e-002$	10.51046	$2.35E-09$	2.09302	$4.81e+000 \pm 5.23e-003$
Band-stop (BS)	PSOwFIPS	17.75	30.28	$2.60e-001 \pm 3.90e-002$	6.70940	$2.06E-06$	2.09302	$5.77e+000 \pm 8.13e-002$
	SaDE	21.17	35.22	$2.04e-001 \pm 5.80e-002$	2.44724	0.02429	2.09302	$6.34e+000 \pm 2.20e-002$
	TLBO	14.01	26.25	$2.30e-001 \pm 7.20e-002$	3.56938	$2.05E-03$	2.09302	$5.34e+000 \pm 3.13e-002$
	DLTLBO	36.07	46.27	$1.60e-001 \pm 4.62e-002$	—	—	—	$5.33e+000 \pm 1.01e-002$

FIGURE 4: Continued.

FIGURE 4: Comparison of the performance curves using different algorithms.

TABLE 5: Optimized coefficients and performance comparison of MES and CPU time.

Algorithm	b_1	b_2	MSE	t -stat	p value	t -value	Time (s)
GA	-1.20001	0.60000	$1.38e-04 \pm 3.37e-04$	1.83706	0.08190	2.09302	$1.41e+01 \pm 2.52e-02$
PSOwFIPS	-1.20000	0.60000	0.0 ± 0.0	NaN	NaN	NaN	$1.50e+01 \pm 3.44e-02$
SaDE	-1.20000	0.60000	0.0 ± 0.0	NaN	NaN	NaN	$1.60e+01 \pm 7.57e-02$
TLBO	-1.20000	0.60000	0.0 ± 0.0	NaN	NaN	NaN	$1.43e+01 \pm 4.70e-02$
DLTLBO	-1.20000	0.60000	0.0 ± 0.0	—	—	—	$1.48e+01 \pm 1.28e-02$

FIGURE 5: Comparison of the performance curves using different algorithms.

Example 2. In the first example (taken from [21, 22, 26]), a second-order IIR filter is estimated by a second-order IIR filter. The unknown plant and the filter had the following transfer functions:

$$\begin{aligned} \text{ideal}H(z) &= \frac{1}{1 - 1.2z^{-1} + 0.6z^{-2}}, \\ \text{actual}H(z) &= \frac{1}{1 + b_1z^{-1} + b_2z^{-2}}. \end{aligned} \quad (10)$$

The input $x(k)$ to the system and the filter is a white sequence. The sequence length used is $N = 100$.

Since the filter order is equal to that of the system, the error surface is unimodal. The best solution should be located at $\{-1.2, 0.6\}$. The cost function is given as shown in (8). All the results are averaged over 20 random runs with randomly chosen initial positions. Table 5 shows the best filter coefficients, the means and standard deviations of MSE, the means and standard deviations of CPU times, and statistical t -test between DLTBLBO and other algorithms. Figure 5 presents the normalized cost function value versus number of cost function evaluations and the magnitude responses of three algorithms.

From the values of MSE in Table 5, one can see that PSOwFIPS, SaDE, TLBO, and DLTBLBO can find the global

minimum. From the values of t -test, it is also seen that DLTBLBO is statistically better than GA. As seen from Table 5 and Figure 5, although DLTBLBO spends more time than TLBO when the number of FEs reaches the given maximum FEs, DLTBLBO has the faster convergence speed than other algorithms. It can be concluded that DLTBLBO has a good performance of the solution accuracy.

Example 3. In the second example (taken from [17, 18, 21, 22, 26]), a second-order IIR filter is estimated by a first-order IIR filter. The unknown plant and the filter had the following transfer functions:

$$\begin{aligned} \text{ideal}H(z) &= \frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}}, \\ \text{actual}H(z) &= \frac{a_0}{1 + b_1z^{-1}}. \end{aligned} \quad (11)$$

The input $x(k)$ to the system and the filter is a Gaussian white noise with the mean of zero and unit variance. The sequence length used is $N = 100$.

Since the reduced order filter is employed for the identification, the error surface of the cost function is multimodal. The error surface has a global minimum at $\{-0.311, -0.906\}$ and a local minimum at $\{0.114, 0.519\}$. The cost function is

TABLE 6: Optimized coefficients and performance comparison of MES and CPU time.

Algorithm	a_0	b_1	MSE	t -stat	p value	t -value	Time (s)
GA	-0.25418	-0.92619	$1.83e-01 \pm 6.74e-05$	2582.69885	$3.76E-54$	2.09302	$1.32e+01 \pm 6.23e-02$
PSOwFIPS	-0.26621	-0.92312	$2.11e-01 \pm 3.69e-02$	8.08353	$1.44E-07$	2.09302	$1.41e+01 \pm 7.63e-02$
SaDE	-0.31900	-0.92058	$2.01e-01 \pm 7.43e-17$	16.75618	$1.44E-12$	2.09302	$1.46e+01 \pm 7.22e-02$
TLBO	-0.26675	-0.91228	$2.07e-01 \pm 4.37e-17$	18.25943	$1.44E-15$	2.09302	$1.33e+01 \pm 4.84e-02$
DLTLBO	-0.36707	-0.89206	$1.44e-01 \pm 5.37e-17$	—	—	—	$1.40e+01 \pm 9.93e-02$

FIGURE 6: Comparison of the performance curves using different algorithms.

given as shown in (8). All the results are averaged over 20 random runs with randomly chosen initial positions. Table 6 shows the comparative results between DLTlBO and other algorithms. Figure 6 presents the normalized cost function value versus number of cost function evaluations and the magnitude responses of three algorithms.

Table 6 shows the best filter coefficients, the means and standard deviations of MSE, and CPU times. It is observed that DLTlBO has the smaller mean value of MSE, while the standard deviations of MSE obtained by SaDE, TLBO, and DLTlBO are approximate. It can be observed from Table 6 that when the number of FEs reaches the given maximum FEs, GA spends less time among all other three algorithms and DLTlBO spends less time than PSOwFIPS and SaDE. It is also seen that the differences of the means of MSE are statistically significant from the values of t -test between DLTlBO and other corresponding algorithms. As seen from Table 6 and Figure 6, the convergence speed of DLTlBO is faster than those of other algorithms.

Example 4. In the third example (taken from [24]), a third-order IIR filter is estimated by a third-order IIR filter. The

unknown plant and the filter had the following transfer functions:

$$\begin{aligned} \text{ideal}H(z) &= \frac{-0.2 + 0.4z^{-1} - 0.5z^{-2}}{1 - 0.6z^{-1} + 0.25z^{-2} - 0.2z^{-3}}, \\ \text{actual}H(z) &= \frac{a_0 + a_1z^{-1} + a_2z^{-2}}{1 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3}}. \end{aligned} \quad (12)$$

The input $x(k)$ to the system and the filter is a Gaussian white noise with the mean of zero and unit variance. The sequence length used is $N = 100$.

Since the filter order is equal to that of the system, the error surface is unimodal. The best solution should be located at $\{-0.2, 0.4, -0.5, -0.6, 0.25, -0.2\}$. The cost function is given as shown in (8). All the results are averaged over 20 random runs with randomly chosen initial positions. Table 7 shows the comparative results between DLTlBO and other algorithms. Figure 7 presents the normalized cost function value versus number of cost function evaluations and the magnitude responses of three algorithms.

As seen from Table 7, the filter coefficients obtained by PSO and DLTlBO are exactly located at the best solution.

TABLE 7: Optimized coefficients and performance comparison of MES and CPU time.

Algorithm	a_0	a_1	a_2	b_1	b_2	b_3	MSE	t -stat	p value	t -value	Time (s)
GA	-0.19546	0.39125	-0.49149	-0.57708	0.22420	-0.18713	$1.47e-03 \pm 7.77e-04$	8.46158	$7.21E-08$	2.09302	$1.44e+01 \pm 9.72e-02$
PSOwFIPS	-0.20000	0.40000	-0.50000	-0.60000	0.25000	-0.20000	$1.54e-17 \pm 1.40e-17$	4.92503	$9.40E-05$	2.09302	$1.53e+01 \pm 1.32e-01$
SaDE	-0.20000	0.40000	-0.50000	-0.60000	0.25000	-0.20000	00 ± 00	-1.70613	0.104277	2.09302	$1.62e+01 \pm 7.40e-02$
TLBO	-0.20000	0.40000	-0.50000	-0.60000	0.25000	-0.20000	$1.20e-32 \pm 2.74e-32$	1.66504	0.11231	2.09302	$1.48e+01 \pm 4.75e-02$
DLTLBO	-0.20000	0.40000	-0.50000	-0.60000	0.25000	-0.20000	$1.89e-33 \pm 4.96e-33$	—	—	—	$1.52e+01 \pm 1.82e-02$

FIGURE 7: Comparison of the performance curves using different algorithms.

TABLE 8: Optimized coefficients and performance comparison of MES and CPU time.

Algorithm	a_0	a_1	b_1	b_2	MSE	t -stat	p value	t -value	Time (s)
GA	-0.41403	-0.06370	-0.26851	-0.52414	$1.32e-02 \pm 1.14e-04$	46.6421	$4.61E-21$	2.09302	$1.42e+01 \pm 1.82e-02$
PSOWFIPS	-0.15324	-0.00954	-1.57619	0.66054	$1.53e-02 \pm 1.57e-03$	9.45093	$1.3E-08$	2.09302	$1.50e+01 \pm 2.84e-02$
SaDE	-0.40588	-0.03978	-0.25151	-0.54818	$1.35e-02 \pm 3.56e-18$	41.65782	$5.82E-20$	2.09302	$1.57e+01 \pm 3.96e-02$
TLBO	-0.37151	-0.04151	-0.20290	-0.61175	$1.50e-02 \pm 3.60e-04$	36.70821	$4.16E-19$	2.09302	$1.44e+01 \pm 8.82e-02$
DTLBO	-0.38470	-0.09199	-0.19445	-0.60222	$1.20e-02 \pm 2.14e-18$	—	—	—	$1.49e+01 \pm 2.42e-02$

Table 7 lists that PSOWFIPS, SaDE, TLBO, and DTLBO have the same best filter coefficients. SaDE offers the best MSE performance and DTLBO outperforms GA, PSOWFIPS, and TLBO. The t -test in terms of the means of MSE between DTLBO and other algorithms indicates that DTLBO is statistically better than the corresponding algorithms except for SaDE. From Table 7 and Figure 7, we can see that the convergence speed of DTLBO is faster than those of other algorithms except SaDE. It can be concluded that DTLBO has a good performance of the solution accuracy, although DTLBO needs more CPU running time when the number of FEs reaches the given maximum FEs.

Example 5. In the fourth example (taken from [17, 18]), a third-order filter is estimated by a second-order IIR filter. The unknown plant and the filter had the following transfer functions:

$$\begin{aligned} \text{ideal}H(z) &= \frac{-0.3 + 0.4z^{-1} - 0.5z^{-2}}{1 - 1.2z^{-1} + 0.5z^{-2} - 0.1z^{-3}}, \\ \text{actual}H(z) &= \frac{a_0 + a_1z^{-1}}{1 + b_1z^{-1} + b_2z^{-2}}. \end{aligned} \quad (13)$$

The input $x(k)$ to the system and the filter is a uniformly distributed white sequence; taking values from $(-0.5, 0.5)$, the SNR = 30 dB. The sequence length used is $N = 100$.

Since the reduced order filter is employed for the identification, the error surface of the cost function is multimodal. The error surface has a global minimum at $\{-0.39, -0.077, -0.22, -0.58\}$. The cost function is given as shown in (8). All the results are averaged over 20 random runs with randomly chosen initial positions. The comparative results between DTLBO and other algorithms are given in Table 8. Figure 8 presents the normalized cost function value versus number of cost function evaluations and the magnitude responses of three algorithms.

As seen from Table 8, DTLBO offers the mean best values and standard deviations of MSE. From the values of t -test between DTLBO and other algorithms, it is also seen that DTLBO is statistically better than the corresponding algorithms. From Figure 8, one can see that convergence speed of TLBO is faster than the other two algorithms. It can also be observed from Table 8 that DTLBO spends the third most times among the five algorithms when the number of FEs reaches the given maximum FEs. It can be concluded that DTLBO could have the best MSE performance among all the

FIGURE 8: Comparison of the performance curves using different algorithms.

five algorithms, although all five algorithms cannot find the global minimum.

6. Conclusion and Future Work

In this paper, TLBO has been extended to DTLBO which uses neighborhood search and differential learning to improve the TLBO algorithm's computation performance. The proposed DTLBO algorithm maintains the diversity of the population effectively and provides more useful information in the iterated process. To demonstrate the effectiveness of our approaches, 24 benchmark functions have been used for simulating and testing. Moreover, DTLBO has been also developed to optimize the coefficients setting of digital IIR filters simultaneously. In the experimental study, DTLBO gained good results on five digital IIR filter design cases even though DTLBO needs to spend much more time when the number of FEs reaches the given maximum FEs. It is shown that the proposed algorithm is potential for function optimization and parameter estimation of digital IIR system in comparison with other approaches.

Further works include research into the following topics. First, neighborhood topological structures and selection of parameters should be investigated to make the DTLBO algorithm more efficient. Secondly, DTLBO of this paper puts emphasis on the use of differential learning, rather than on analyzing the influence of different learning formations of differential evolution on DTLBO. The latter problem would be interesting to investigate in future research. Thirdly, DTLBO could be used in a wide variety of filter design applications. Tradeoff between various parameters of the filter can lead to designing different kinds of filters according to different requirements in various kinds of applications.

Finally, it is expected that DTLBO will be used in other real-world optimization problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was partially supported by National Natural Science Foundation of China (61100173, 61100009, 61272283, 61304082, and 61572224). This work is also supported by Natural Science Foundation of Anhui Province (Grant no. 1308085MF82) and Major Project of Natural Science Research in Anhui Province under Grant KJ2015ZD36.

References

- [1] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.
- [2] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems," *Information Sciences*, vol. 183, no. 1, pp. 1–15, 2012.
- [3] V. Toğan, "Design of planar steel frames using teaching-learning based optimization," *Engineering Structures*, vol. 34, pp. 225–232, 2012.
- [4] R. V. Rao and V. Patel, "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 3, no. 4, pp. 535–560, 2012.

- [5] S. O. Degertekin and M. S. Hayalioglu, "Sizing truss structures using teaching-learning-based optimization," *Computers and Structures*, vol. 119, pp. 177–188, 2013.
- [6] R. V. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Scientia Iranica*, vol. 20, no. 3, pp. 710–720, 2013.
- [7] T. Niknam, F. Golestaneh, and M. S. Sadeghi, " θ -Multiobjective teaching-learning-based optimization for dynamic economic emission dispatch," *IEEE Systems Journal*, vol. 6, no. 2, pp. 341–352, 2012.
- [8] F. Zou, L. Wang, X. Hei, D. Chen, and D. Yang, "Teaching-learning-based optimization with dynamic group strategy for global optimization," *Information Sciences*, vol. 273, pp. 112–131, 2014.
- [9] L. Wang, F. Zou, X. Hei, D. Yang, D. Chen, and Q. Jiang, "An improved teaching-learning-based optimization with neighborhood search for applications of ANN," *Neurocomputing*, vol. 143, pp. 231–247, 2014.
- [10] G. Waghmare, "Comments on 'a note on teaching-learning-based optimization algorithm,'" *Information Sciences*, vol. 229, pp. 159–169, 2013.
- [11] J. J. Shynk, "Adaptive IIR filtering," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 6, no. 2, pp. 4–21, 1989.
- [12] M. Radenkovic and T. Bose, "Adaptive IIR filtering of nonstationary signals," *Signal Processing*, vol. 81, no. 1, pp. 183–195, 2001.
- [13] S. Stearns, "Error surfaces of recursive adaptive filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 3, pp. 763–766, 1981.
- [14] K.-S. Tang, K.-F. Man, S. Kwong, and Z.-F. Liu, "Design and optimization of IIR filter structure using hierarchical genetic algorithms," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 3, pp. 481–487, 1998.
- [15] J.-T. Tsai, J.-H. Chou, and T.-K. Liu, "Optimal design of digital IIR filters by using hybrid Taguchi genetic algorithm," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 3, pp. 867–879, 2006.
- [16] Y. Yang and X. Yu, "Cooperative coevolutionary genetic algorithm for digital IIR filter design," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1311–1318, 2007.
- [17] S. Chen, R. Istepanian, and B. L. Luk, "Digital IIR filter design using adaptive simulated annealing," *Digital Signal Processing*, vol. 11, no. 3, pp. 241–251, 2001.
- [18] A. Kalinli and N. Karaboga, "A new method for adaptive IIR filter design based on tabu search algorithm," *AEU—International Journal of Electronics and Communications*, vol. 59, no. 2, pp. 111–117, 2005.
- [19] R. Storn, "Designing nonstandard filters with differential evolution," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 103–106, 2005.
- [20] N. Karaboga and B. Çetinkaya, "A simple evolutionary algorithm for IIR filter design: differential evolution algorithm," in *Proceedings of the IEEE 13th Signal Processing and Communications Applications Conference (SIU '05)*, pp. 183–186, IEEE, May 2005.
- [21] S. Yuenyong and A. Nishihara, "A hybrid gradient-based and differential evolution algorithm for infinite impulse response adaptive filtering," *International Journal of Adaptive Control & Signal Processing*, vol. 28, no. 10, pp. 1054–1064, 2014.
- [22] N. Karaboga, A. Kalinli, and D. Karaboga, "Designing digital IIR filters using ant colony optimisation algorithm," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 3, pp. 301–309, 2004.
- [23] N. Karaboga, "A new design method based on artificial bee colony algorithm for digital IIR filters," *Journal of the Franklin Institute*, vol. 346, no. 4, pp. 328–348, 2009.
- [24] S. Chen and B. L. Luk, "Digital IIR filter design using particle swarm optimisation," *International Journal of Modelling, Identification & Control*, vol. 9, no. 4, pp. 327–335, 2010.
- [25] A. Sarangi, R. K. Mahapatra, and S. P. Panigrahi, "DEPSO and PSO-QI in digital filter design," *Expert Systems with Applications*, vol. 38, no. 9, pp. 10966–10973, 2011.
- [26] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Filter modeling using gravitational search algorithm," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 117–122, 2011.
- [27] S. K. Saha, R. Kar, D. Mandal, and S. Ghoshal, "Gravitation search algorithm: application to the optimal IIR filter design," *Journal of King Saud University—Engineering Sciences*, vol. 26, no. 1, pp. 69–81, 2014.
- [28] A. P. Patwardhan, R. Patidar, and N. V. George, "On a cuckoo search optimization approach towards feedback system identification," *Digital Signal Processing*, vol. 32, pp. 156–163, 2014.
- [29] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [30] L. Wang, F. Zou, X. Hei et al., "A hybridization of teaching-learning-based optimization and differential evolution for chaotic time series prediction," *Neural Computing and Applications*, vol. 25, no. 6, pp. 1407–1422, 2014.
- [31] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [32] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [33] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
- [34] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '03)*, pp. 174–181, April 2003.
- [35] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, Pearson Prentice Hall, 2007.
- [36] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.