A New Algorithm (ESA-DE) for Designing FIR Digital Filters

LiBao Deng(✉), Haili Sun, and Lili Zhang

Harbin Institute of Technology, Weihai, Shandong, China
denglibao_paper@163.com

Abstract. This paper proposes a new algorithm called (ESA-DE) for software-designed realization of finite impulse response (FIR) filters. The determination of the optimal order of the filter is always a confusing problem. In this paper, we will confirm the optimal order of the filter. Differential evolution has obtained widely concern as one of the most promising intelligent optimization algorithms in the field of artificial intelligence. The new algorithm (ESA-DE) uses a new mechanism called elite guide with weight and two new self-adaptive parameter distribution techniques to enhance algorithm's diversity and convergence. Experiments prove that the new algorithm is at least superior to the basic DE and its variants jDE and ODE. The proposed algorithm is applied to solve practical problem that it is to design FIR digital filters, which overcomes the shortcomings of the traditional designing method. In addition, the practical problem about design for digital filter demonstrates that the ESA-DE has a better performance compared with the other three algorithms.

Keywords: FIR digital filters · Artificial intelligence · Differential evolution (DE)

1 Introduction

FIR is a kind of the digital filter [1] which is widely used for stability, amplitude's randomicity and limited sampling value. The window function method [2,3], optimal least-squares method (LSM) [4,5]and other traditional methods have problems on determining the boundary frequency of passband and stopband. The traditional optimization algorithms only pay attention to local information which can't meet the requirement of high precision of practical engineering problems because it is difficult to jump out of local optimum. Intelligent optimization methods with global optimization ability gradually emerge, including

This work was supported in part by National Natural Science Foundation of China (61401121).

M. Jia et al. (Eds.): WiSATS 2019, LNICST 280, pp. 640–652, 2019.
https://doi.org/10.1007/978-3-030-19153-5_63

genetic algorithm (GA) [6], particle swarm algorithm (PSO) [7] and differential evolutionary algorithm (DE) [8], etc. These algorithms simulate the evolutionary process of natural organisms or the behavior of social groups of organisms to form a scientific calculation method called artificial life computation. Scholars continuously devote to the artificial life and design research of digital filter with artificial life calculation.They put forward a new optimization algorithm [9,10] and various technologies [5,11–13] to design digital filter and achieve good optimization results.

The layout of the rest of the paper is as follows: Sect. 2 gives a brief introduction about principle of FIR digital filter and model establishment. Section 3 reviews the related works on the traditional DE. Section 4 elaborates the new algorithm with various strategies and comparative experiments results of several algorithms. Section 5 demonstrates design process of low-pass FIR digital filter. Finally, a conclusion is made in Sect. 6.

2 FIR Digital Filter Principle

In this section, principle and model of FIR digital filter is introduced.

2.1 Principle of FIR Digital Filter

The optimal design of digital filter is an important part of signal processing. It can be realized by hardware [14] and software while software implementation has become a trend of filter design. Both can process the digital discrete signal and change the frequency response or waveform of the input signal. They can prevent the interference signal from outputting and then output the required signal. FIR digital filter has a wide range of applications including speech recognition, image processing, communication , military target navigation [15,16], the trend of the economic market forecast, energy distribution of power system [17], and intelligent robot system automatically detecting [18] etc.

2.2 Establishment of FIR Digital Filter Model

We designed a linear phase FIR digital filter [18] and selected finite impact response $h(n)(1 \leqslant n \leqslant N-1$, N represents the filter order). The system function $H(z)$ is as follows:

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} \tag{1}$$

The frequency response function corresponding to the filter is as follows:

$$H(e^{jw}) = \sum_{n=0}^{N-1} h(n) e^{-jwn} \tag{2}$$

The filter has a linear phase and obtains four different amplitude responses in the frequency domain. One odd number is listed here ($\theta(w)$ is the phase frequency response) (Table 1).

Table 1. Parameter correspondence of amplitude responses

N = odd	$H(w)$ and $\theta(w)$	Relation of $H(w)$ and $h(n)$	The filter type
$h(n)$ = even symmetry	$\sum_{n=0}^{\frac{N-1}{2}} h_1(n)cos(wn)$ $-w\frac{N-1}{2}$	$h_1(0) = h(\frac{N-1}{2})$ $h_1(n) = 2h(\frac{N-1}{2} + n)$ $n = 1, 2, ..., \frac{N-1}{2}$	(1) Low pass (2) High pass (3) Band pass (4) Stop band

3 Classical DE

DE is a stochastic optimization algorithm based on the number of population NP, which searchs the optimal solution in the D dimensional space. We use $X_{i,G} = [x_{i,1,G}, x_{i,2,G}, ..., x_{i,D,G}]$ to represent the ith candidate solution in generation G, where $i = 1, 2, ..., NP$. For the classical DE [19], there are four following operations:initialization, mutation, crossover, and selection.

3.1 Initialization

We initialize the candidate solution between lower limit $X_{i,G} = [x_{1,min}, x_{2,min}, ..., x_{D,min}]$ and high limit $X_{i,G} = [x_{1,max}, x_{2,max}, ..., x_{D,max}]$. The following equation is used to initialize individuals in the initial population :

$$x_{i,j} = x_{j,min} + rand_{i,j}(0, 1) \cdot (x_{j,max} - x_{j,min}) \tag{3}$$

where $x_{j,min}$ and $x_{j,max}$ denote the lower and upper bound, It's a constant between [0, 1].

3.2 Mutation

Mutation operation is the part of DE algorithm. Each target vector $X_{i,G}$ is viewed as the base vector once. And then, two target vectors $X_{i2,G}$ and $X_{i3,G}$ are selected from the population. The difference vector is generated. The difference vector is weighted and added to the basis vector. The mutation vector $V_{i,G} = [v_{i,1,G}, v_{i,2,G}, ..., v_{i,D,G}]$ is obtained. The basic variation strategy:

$$V_{i,G} = X_{i1,G} + F \cdot (X_{i2,G} - X_{i3,G}) \tag{4}$$

where $i = 1, 2, ..., NP$. index $i1, i2, i3 \in [1, NP]$, which is different from each other.

3.3 Crossover

The crossover operation is to mix the components of the target vector $X_{i,G}$ and the mutation vector $V_{i,G}$ and get the trial vector $U_{i,G} = [u_{i,1,G}, u_{i,2,G}, ..., u_{i,D,G}]$ which can be generated by the following formula:

$$u_{i,j,G} = \begin{cases} v_{i,j,G} & \text{if } rand_{i,j}(0,1) \leqslant CR \text{ or } j = j_{rand} \\ x_{i,j,G} & \text{otherwise} \end{cases} \tag{5}$$

where $i = 1, 2, ..., NP$. $rand_j(0,1)$ is randomly chosen from 0 to 1 for each j and each i according to a uniform distribution. the crossover factor CR is a constant between [0, 1], which determines how many components of the trial vector from the mutation vector and affects the evolutionary speed of the algorithm. j_{rand} that is a integer keeps that one different parameter exists between the target vector $X_{i,G}$ and its trial vector $U_{i,G}$ at least.

3.4 Selection

The selection operation forms a new population by selecting the trial vector and corresponding target vector. According to their fitness values $f(\cdot)$, The vector with having the best fitness value is preserved. The implementation is as follows:

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{if } f(U_{i,G}) \leqslant f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \tag{6}$$

where $f(\cdot)$ represents the fitness value of the function. In general, we optimize the minimum value of the problem.

3.5 Condition of Algorithm Termination

There are two kinds of circumstances to make algorithm to end. On the one hand, the number of iteration has reached maximum number of iteration, on the other hand, the error $f_{actual}(\cdot) - f(\cdot)$ between the ideal function value $f(\cdot)$ and actual output function value $f_{actual}(\cdot)$ is not more than 10^{-8}.

4 ESA-DE

The convergence speed of the algorithm is accelerated by using strategy of elite guidance with weight. We use adaptive F and CR to balance the exploration and exploitation. The description is as follows.

4.1 Strategy of Elite Guide with Weight

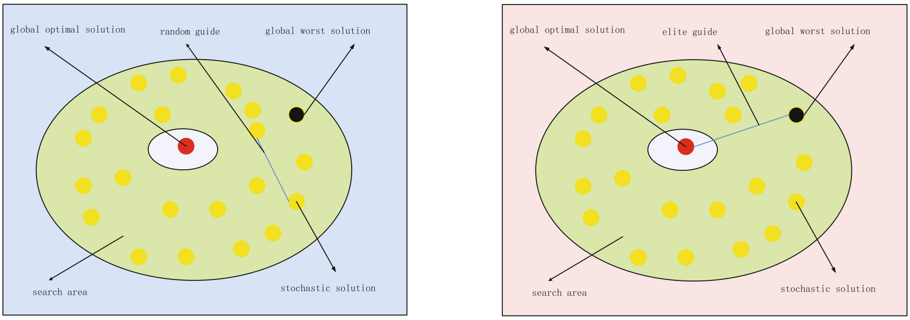
Variation strategy [23,24] has a direct impact on the performance of the algorithm. We aim to better find the global optimal solution, in the first place, population diversity should be maintained to avoid the algorithm falling into the local optimum. In the second place, the convergence speed should be guaranteed. Direction of the improvement which based on the above content that is strategy of elite guide with weight is put forward in this paper. It will retain the original difference vector of DE algorithm to ensure the population diversity. What's more, it will join a new difference vector and let the elite guide the worst evolve together to speed up the algorithm convergence through diagram Figs. 1a and 1b. In addition, the weight w is added to control the participation degree of two difference vectors to ensure the accuracy of the solution. The strategy equation is as follows:

$$X_1 = X_{r1} - X_{r2} \tag{7}$$

$$X_2 = X_{best} - X_{worst} \tag{8}$$

$$V = X_i + w \cdot F \cdot X_1 + (1 - w) \cdot F \cdot X_2 \tag{9}$$

where X_1 and X_2 represent two difference vectors, X_i , X_{r1} and X_{r2} are chosen at random from a population, $i \neq r1 \neq r2$; X_{best} is the best individual in the population, X_{worst} is the worst individual in the population; $w = 0.9$; F is the scale factor, which controls the magnitude of the difference vector between [0,1].



(a) *random-guide strategy* **(b)** *elite-guide strategy*

Fig. 1. Mapping of logistic and sinusoidal

4.2 Parameter Adaptation

Parameter setting [25–28] is another important factor to affect the performance of the algorithm. The two main parameters of the DE algorithm are F and CR. Changeable parameters can improve the performance of the algorithm so we let the parameters change in each generation to adapt to the evolution of the algorithm. In order to improve the convergence speed of the algorithm and carry out local accurate search. We let F keep large to ensure the diversity of the population at an early stage. In the later stage of the algorithm evolution, F decreases gradually. Similarly, CR was large to ensure the convergence speed in the early stage. In the later stage of evolution, CR was small to ensure the diversity of population. Based on the above ideas, the distribution of F and CR is as follows:

$$F = F_{max} - \frac{F_{max} - F_{min}}{G_{max}} \cdot G \tag{10}$$

$$CR = \frac{CR_{min}}{1 + (\frac{CR_{min}}{CR_{max}} - 1) \cdot e^{-G}} \tag{11}$$

where G_{max} is maximum number of iteration, G is current iteration number, F_{max} and F_{min} are maximum and minimum value of variation factors, they are set to 0.9 and 0.3, CR_{max} and CR_{min} are maximum and minimum value of crossover factors, they are set to 0.8 and 0.5, algorithm flow is displayed algorithm 1.

Algorithm 1. Pseudocode of OTWDE algorithm

Begin
    $G\_max = 1000$; $NP = 100$;$w = 0.9$;
    $F\_min = 0.9$; $F\_max = 0.3$;$CR\_min = 0.8$; $CR\_max = 0.5$;
  Inilization
  Generate a random initial population $P_0 = [X_1, X_2, ..., X_{NP}]$
  while $G < G\_max$ **do**
   for $i = 1 : NP$ **do**
  According to 10 and 11 to update F and CR
  According to 9 to generate $V_{i,G}$
  According to 5 to generate $U_{i,G}$
  According to 6 to select the next generation $X_{i,G+1}$
    if $f(U_{i,j,g}) \leq f(X_{i,j,g})$ **then**
     $X_{i,j,G+1} = U_{i,j,G}$;
    else
     $X_{i,j,G+1} = X_{i,j,G}$
    end if
   end for
  end while

4.3 Test Function

For the evaluation of algorithm performance, there is no unified evaluation criterion in academia while the performance comparison of optimization algorithm is usually based on some typical problems called Benchmark [29,30]. In this paper, 6 classic universal test functions are selected, High Conditioned Elliptic Function f(1), Rastrigin Function f(2), Rosenbrock Function f(3), Cigar Function f(4), Griewank Function f(5), Schaffer Function f(6), respectively. The specific description is given Table 2 (Fig. 2).

4.4 Simulation Result

The performance of DE, jDE, ODE and ESA-DE algorithm is evaluated by the above 6 common test functions. The four algorithms run 50 times independently. F and CR are 0.5 and 0.3 for standard DE algorithm, respectively; The parameter setting of jDE and ODE is consistent with the original algorithm; For ESA-DE algorithm, $F_{max} = 0.9$, $F_{min} = 0.3$, $CR_{max} = 0.8$,$CR_{min} = 0.5$. The maximum number of iteration of the four algorithms is $G_{max} = 1000$. The population size is set $NP = 100$. We obtained the performance evaluation criterion of

Table 2. Specific description of test functions

Function name	Seach range	Optimum	Dimension
f_1	$[-30, 30]^D$	0	50
f_2	$[-5.12, 5.12]^D$	0	20
f_3	$[-30, 30]^D$	0	20
f_4	$[-30, 30]^D$	0	50
f_5	$[-500, 500]^D$	0	20
f_6	$[-100, 100]^D$	0	20

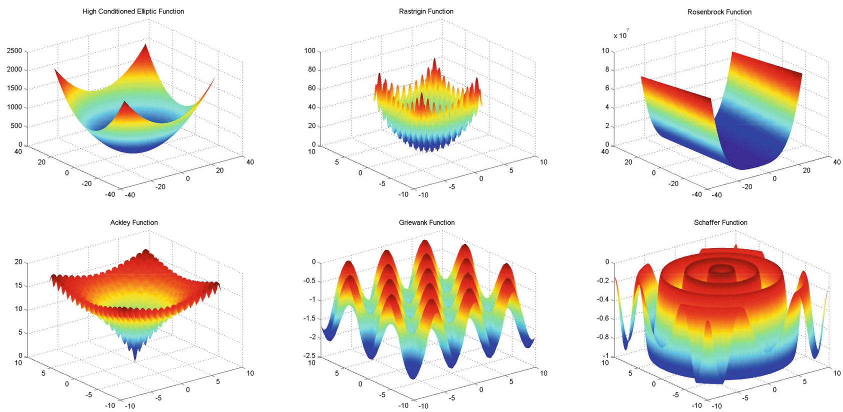


Fig. 2. 3-dimensional images of 2-dimensional functions of 6 test functions

test function: average error value of fitness (mean), standard deviation (std) and the success rate of the solution (SR) ($SR = SN/S$,SN is amounts that we find the optimal value among 50 times, S is the total number of 50 runs). Table 3 represents the data simulation results. Figure 3 is convergence figures of test functions. We see Table 3, ESA-DE is less than mean and standard error value of other algorithms for the mean and standard error value. The success rate of solution is also higher than other algorithms. It can illustrate that ESA-DE has higher robustness and precision about solution. We can conclude that ESA-DE converges earlier than other algorithms in earlier generation which means that ESA-DE converges faster From the convergence simulation diagram of Fig. 3. In conclusion, the improved algorithm is superior to the standard DE algorithm, jDE and ODE.

Table 3. Mean, standard deviation value and SR of ESA-DE, DE, jDE and ODE over 50 independent runs

Function	Algorithm	Mean	Std	SR
$f_1(50)$	ESA-DE	1.19E−14	8.46E−15	100%
	DE	1.57E−09	1.15E−09	100%
	jDE	6.94E−11	9.30E−11	100%
	ODE	1.55E−12	6.01E−13	100%
$f_2(30)$	ESA-DE	5.00E−10	3.50E−10	100%
	DE	9.05E+01	7.10E+00	0%
	jDE	5.70E+02	2.61E+01	0%
	ODE	2.38E+01	4.48E+00	13%
$f_3(30)$	ESA-DE	1.44E−02	1.22E−02	15%
	DE	3.67E+01	1.55E+01	0%
	jDE	3.28E+02	1.81E+02	0%
	ODE	1.23E−05	1.70E−05	40%
$f_4(50)$	ESA-DE	1.64E−16	1.00E−16	100%
	DE	1.49E−04	6.97E−05	0%
	jDE	4.92E−09	3.44E−09	90%
	ODE	5.73E−09	4.05E−08	87%
$f_5(30)$	ESA-DE	2.32E−13	2.18E−13	100%
	DE	2.28E+00	7.39E+00	0%
	jDE	3.78E−07	4.54E−06	40%
	ODE	5.69E−08	7.24E−09	93%
$f_6(30)$	ESA-DE	2.47E−08	1.10E−08	93%
	DE	8.93E+01	3.30E−01	0%
	jDE	1.44E−05	1.40E−06	30%
	ODE	1.85E−03	3.82E−04	0%

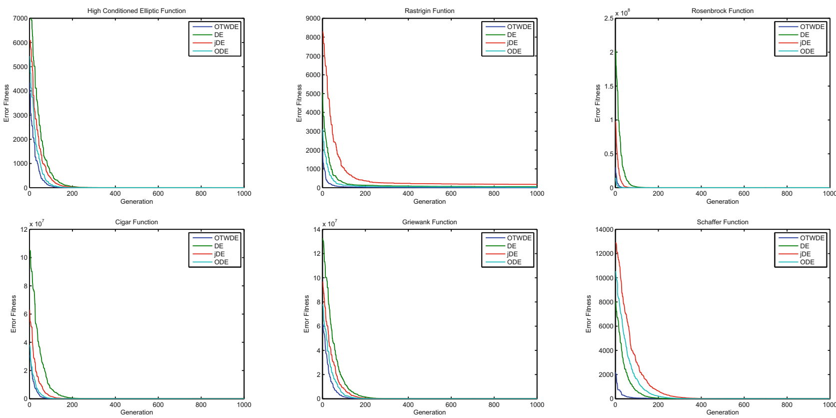


Fig. 3. Convergence figures of 6 test functions

5 Design of Low Pass FIR Digital Filter

The software-designed filter is a popular topic for digital signal processing of artificial intelligence in recent years. DE algorithm is booming in recent years. It has found application in various fields and achieved good results. Besides relying on theory, The test of algorithm design needs validation of test problems so we use DE to design filter.

5.1 Establish the Fitness Function Model

The difference evolution is used to evaluate individual's advantages and disadvantages through fitness value so you must choose the corresponding fitness function E according to the actual problem. The minimum square error optimization design of the FIR digital filter is required at the discrete frequency point w_i,and $i = 1, ..., M$. FIR digital filter and the ideal FIR digital filter are $|H(e^{jw_i})|$ and $|H_d(e^{jw_i})|$ respectively, The fitness function of the algorithm is shown as follows:

$$E = \sum_{i=1}^{M} [|H(e^{jw_i})| - |H_d(e^{jw_i})|]^2 = \sum_{i=1}^{M} [| \sum_{n=0}^{N-1} h(n)(e^{-jw_i})| - |H(e^{jw_i})|]^2 \quad (12)$$

where the smaller E, it shows that optimized parameters are better and the filter has a better performance.

5.2 Parameter Coding

We designed a FIR low-pass digital filter that N is odd which is based on the corresponding process of coefficient $h(n)$ and $h_1(n)$. We use the symmetry of $h_1(n)$ to get all coefficients of filter. $h_1(n)$ is encoded by matrix as follows:

$$\begin{bmatrix} H(w_1) \\ \cdots \\ H(w_M) \end{bmatrix} = \begin{bmatrix} 1 & cos(w_1) & \cdots & cos(\frac{N-1}{2} w_1) \\ \cdots & \cdots & \ddots & \cdots \\ 1 & cos(w_1) & \cdots & cos(\frac{N-1}{2} w_M) \end{bmatrix} \begin{bmatrix} h_1(0) \\ \cdots \\ h_1(\frac{N-1}{2}) \end{bmatrix} \quad (13)$$

where the matrix equation is written as $H = A \cdot h$. H represents sampling value vector of the frequency domain. A coefficient matrix. h represents the filter coefficient vector. The fitness function of the algorithm can be rewritten as:

$$E = (|A \cdot h_1 - H_d|)^{'} (|A \cdot h_1 - H_d|) \quad (14)$$

where the individual that we encode is a $\frac{N-1}{2}$ dimensional variable.

5.3 Requirements for Technical Indicators

Low-pass filter means that some signals below the cut-off frequency can pass smoothly while those above the cut-off frequency can't pass. The designed FIR low-pass digital filter is designed as follows:

$$H_d(e^{jw}) = \begin{cases} 1 & 0 \leqslant w < 0.4\pi \\ 0 & 0.5\pi \leqslant w < \pi \end{cases} \quad (15)$$

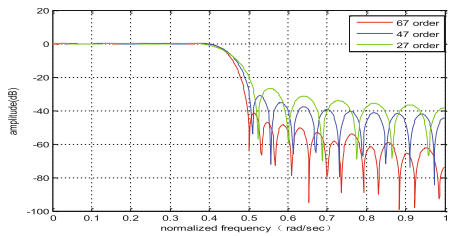


Fig. 4. Situation of order increasing

where the optimized filter parameter values are reserved with 5 decimal places.

5.4 Design Flow

The main steps of designing a linear phase digital filter using ESA-DE algorithm are as follows: (1) give the technical index of designed linear phase digital filter; (2) the parameters of ESA-DE algorithm are set according to the design requirement of FIR digital filter; (3) the population was randomly initialized within the parameter range and the fitness value was calculated; (4) carry out mutation, crossover and selection for each target individual; (5) repeat (4) until the end condition of the algorithm is satisfied and finally output FIR digital filter parameters.

5.5 Result Analysis

We first determine the optimal order of the digital low pass filter. We use four algorithms to design the filter according to the optimal order. Finally we prove the superiority of ESA-DE algorithm from the practical problems.

Determination of the Optimal Order. The order N will affect the approximation of the designed filter to the ideal filter. The distortion degree of sampling will be smaller for the greater N. The problem with great N is that complexity increases gradually. Even if the narrow pulse is wide, filtering effect will decline. This paper determines the optimal order size. We first applied ESA-DE to design FIR low pass digital filter. Cut-off frequecy of passband is set to 0.4π. Cut-off frequency of stopband is set to 0.5π. We select 64 sampling points. The simulation results are shown in Figs. 4 and 5.

Filter Design of Four Algorithms. Four algorithms are used to design FIR low-pass digital filters. The parameters encoding, requirements for technical indicators and design flow are the same as the above design method. It proves that the improved algorithm is better than standard DE algorithm, jDE and ODE through examples of practical problem. According to the above simulation results, the order of the filter is set to 67. The simulation results are shown in Fig. 6.

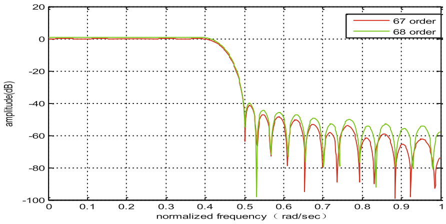


Fig. 5. Situation of order determination

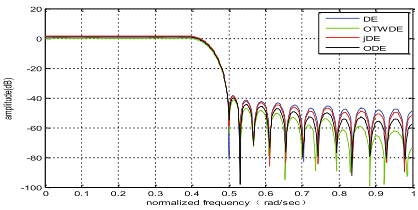


Fig. 6. Design situation of four algorithms

Filter Design of Four Algorithms. As can be seen from the simulation results, there is no significant difference in the optimization effect of several algorithms in the passband and transition zone. The passband ripple is basically 0, indicating that the robustness of the DE algorithm is very good. On the side of stopband attenuation, the designed filter attenuation amplitude for ESA-DE is greater than 30 dB. Attenuation amplitude of the designed filter for DE algorithm and jDE are all greater than 10 dB. The designed filter attenuation amplitude for the ODE is greater than 20 dB. It can also explain that ESA-DE is superior to the above three kinds of algorithms.

6 Conclusion

In this paper, the basic DE algorithm is improved. The improvements including two aspects: (1) Elite guiding strategy is used to make the best individual guide the worst individual (2) adaptive parameter mechanism is utilized in the different stages of evolution algorithm. Importantly, The optimal order of filter is determined. Posteriorly, basic DE algorithm, jDE and ODE are also used to design filters. The comparison experiments show that the proposed algorithm performs better than the above three algorithms in solving the practical problem from the opposite side. In the future, we will use the improved algorithm to try to solve magnitude-error and phase-error constrained optimal designs of finite impulse response (FIR) filter problem.

References

1. Haridas, N., Elias, E.: Low-complexity technique to get arbitrary variation in the bandwidth of a digital FIR filter. IET Sign. Process. **11**(4), 372–377 (2017)
2. Kumar, S., Mehra, R., Chandni.: Implementation and designing of FIR filters using Kaiser window for de-noising of electrocardiogram signals on FPGA. In: IEEE Power India International Conference, pp. 1–6. IEEE (2016)
3. Das, P., Naskar, S.K., Patra, S.N.: An approach to enhance performance of Kaiser window based filter. In: International Conference on Research in Computational Intelligence & Communication Networks, pp. 256–261. IEEE (2017)
4. Eghbali, A., et al.: Optimal least-squares FIR digital filters for compensation of chromatic dispersion in digital coherent optical receivers. J. Lightwave Technol. **32**(8), 1449–1456 (2014)
5. Yi, H., et al.: A new constraint model for optimal design of constrained FIR digital filters. In: Chinese Control Conference, pp. 5566–5571. IEEE (2017)
6. Nyathi, T., Pillay, N.: Comparison of a genetic algorithm to grammatical evolution for automated design of genetic programming classification algorithms. Expert Syst. Appl. **104**, 213–234 (2018)
7. Nobile, M.S., et al.: Fuzzy self-tuning PSO: a settings-free algorithm for global optimization. Swarm Evol. Comput. **39**, 70–85 (2017)
8. Price, K., Price, K.: Differential Evolution a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Kluwer Academic Publishers, Dordrecht (1997)
9. Jiang, A., Kwan, H.K.: WLS design of sparse FIR digital filters. IEEE Trans. Circuits Syst. I Regul. Pap. **60**(1), 125–135 (2013)
10. Das, P., Naskar, S.K., Patra, S.N.: Adaptive global best steered Cuckoo search algorithm for FIR filter design. In: International Conference on Research in Computational Intelligence & Communication Networks, pp. 15–20 (2017)
11. Shiung, D., Yang, Y.Y., Yang, C.S.: Improving FIR filters by using cascade techniques tips & tricks. IEEE Sign. Process. Mag. **33**(3), 108–114 (2016)
12. Mason, J.S., Chit, N.N.: New approach to the design of FIR digital filters. In: IEE Proceedings G - Electronic Circuits and Systems, vol. 134, no. 4, pp. 167–180 (1987)
13. Bose, A., Chandra, A.: Conditional differential coefficients method for the realization of powers-of-two FIR filter. IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst. **PP**(99), 1 (2018)
14. Chandra, A., Chattopadhyay, S.: Design of hardware efficient FIR filter: a review of the state-of-the-art approaches. Eng. Sci. Technol. Int. J. **19**(1), 212–226 (2016)
15. Tajima, S., Sencer, B., Shamoto, E.: Accurate interpolation of machining tool-paths based on FIR filtering. Precis. Eng. **52**, 332–344 (2018)
16. Eren, L., Akar, M., Devaney, M.J.: Motor current signature analysis via four-channel FIR filter banks. Measurement **89**, 322–327 (2016)
17. Xiong, H., Li, D.: Nuclear reactor doubling time calculation using FIR filter. Energy Procedia **39**, 3–11 (2013)
18. Pak, J.M., et al.: Maximum likelihood FIR filter for visual object tracking. Neurocomputing **216**(C), 543–553 (2016)
19. Ozer, A.B.: CIDE: chaotically initialized differential evolution. Expert Syst. Appl. **37**(6), 4632–4641 (2010)
20. May, R.M.: Simple mathematical models with very complicated dynamics. Nature **261**(5560), 459–467 (1976)

21. Peitgen, H.O.: Chaos and fractals: new frontiers of science. Math. Gaz. **79**(484), 241–255 (2004)
22. He, Y., et al.: Differential evolution algorithm combined with chaotic pattern search. Kybernetika-Praha **46**(4), 684–696 (2010)
23. Brest, J., Maučec, M.S.: Population size reduction for the differential evolution algorithm. Appl. Intell. **29**(3), 228–247 (2008)
24. Yang, Q., et al.: Multimodal estimation of distribution algorithms. IEEE Trans. Cybern. **47**(3), 636–650 (2017)
25. Ali, M.M.: Differential evolution with generalized differentials. J. Comput. Appl. Math. **235**(8), 2205–2216 (2011)
26. Weber, M., Neri, F., Tirronen, V.: A study on scale factor in distributed differential evolution. Inf. Sci. **181**(12), 2488–2511 (2011)
27. Zou, D., et al.: An improved differential evolution algorithm for the task assignment problem. Eng. Appl. Artif. Intell. **24**(4), 616–624 (2011)
28. Fu, H., Xu, J., Xu, J.: A Self-adaptive Differential Evolution Algorithm for Binary CSPs. Pergamon Press, Inc., Oxford (2011)
29. Zamuda, A., Brest, J., Mezura-Montes, E.: Structured population size reduction differential evolution with multiple mutation strategies on CEC 2013 real parameter optimization. In: 2013 IEEE Congress on Evolutionary Computation, Cancun, 2013, pp. 1925–1931 (2013). https://doi.org/10.1109/CEC.2013.6557794
30. Liang, J.J., Qu, B.Y., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization (2013)