A State Evaluation Adaptive Differential Evolution Algorithm for FIR Filter Design

Yong WANG[1], Zhaosheng TENG[1], He WEN[1], Jianmin LI[1], Radek MARTINEK[2]

[1]College of Electrical and Information Engineering, Hunan University, Changsha, China
[2]Department of Cybernetics and Biomedical Engineering, Faculty of Electrical Engineering and Computer Science, VSB–Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava, Czech Republic

wangyong1179@163.com, tengzs@126.com, he_wen82@126.com,
lijianmindzyx@163.com, radek.martinek@vsb.cz

Abstract. *Due to conventional differential evolution algorithm is often trapped in local optima and premature convergence in high dimensional optimization problems, a State Evaluation Adaptive Differential Evolution algorithm (SEADE) is proposed in this paper. By using independent scale factor on each dimension of optimization problem, and evaluating the distribution of population on each dimension, the SEADE correct the control parameters adaptively. External archive and a moving window evaluation mechanism on evolution state are introduced in SEADE to detect whether the evolution is stagnation or not, and with the help of opposition-based population, the algorithm can jump out of local optima basins. The results of experiments on several benchmarks show that the proposed algorithm is capable of improving the search performance of high dimensional optimization problems. And it is more efficient in design FIR digital filter using SEADE than conventional method like Parks-McClellan algorithm.*

Keywords

Differential Evolution (DE), external archive, FIR filter, opposition-based population, state evaluation.

1. Introduction

FIR digital filter is an important component of digital signal processing system [1]. With the characteristics of system stability, easy to achieve linear phase, allowing to design of multi passband/stopband and easy to implement in hardware, FIR digital filter has been widely used in communications, voice and image processing, radar, biomedical systems, consumer electronics system, seismic exploration and other fields [2]. Traditional design methods of FIR digital filter include window function method [3], frequency sampling method [4] and uniform approximation method [5]. Among them, the frequency sampling method and the window function method are simple, but it is difficult to accurately determine boundary frequency of passband and stopband of the filter. Uniform approximation method, such as Parks-McClellan algorithm [6], can obtain better passband and stopband performance, and can accurately specify the passband and stopband edge, but the amplitude error relative value of frequency band is specified by the weighting function rather than by the deviation of FIR digital filter.

In recent years, various computational intelligence algorithms have begun to be applied to the optimal design of FIR digital filter and various unbiased state-space filters, and achieved good results [7] and [8]. Differential Evolution (DE for short) algorithm, which is proposed by Storn and Price [9], is a simple and effective random optimization algorithm based on population. Due to its excellent extendibility and versatility, DE has been widely used in various fields. However, similar to other evolutionary algorithms, DE is easy to fall into local optima and premature convergence in solving high dimensional optimization problems. In order to overcome the shortcomings of DE, researchers have proposed a variety of improved algorithms based on DE. The improvement is focused on the parameter adjustment and control of DE [10], and there are three main types of improvements [11] and [12]:

- Modify the control parameters of DE using deterministic parameter control rules, such as the linear change of control parameters with the evolutionary iterations [13].

- Modify and adjust the control parameters of DE according to the feedback of evolutionary search [14], [15] and [16].

- Self-adaptive improvement of DE, in which the control parameters are encoded into the population, and evolve together with the population, such as the self-adaptive DE algorithm(jDE) [17], SaDE [18] and ISAMODE-CMA [19].

In this paper, we present a State Evaluation Adaptive Differential Evolution algorithm (SEADE), aiming at the shortcomings of DEs in solving high dimensional optimization problems. SEADE correct the control parameters using the feedback of the state of population and evolution, and jump out of local optima basins through external archive and opposition-based population. The simulation results show that the SEADE applied to the optimal design of FIR digital filter is superior to other optimization algorithms.

2. Differential Evolution Algorithm

2.1. Optimization Problem Model

Optimization problems in practical scientific research and engineering applications can be formulated as minimization problems which can be expressed in general:

$$\min f(\vec{x}), \qquad (1)$$

where $f : \Omega \subseteq \Re^D \longrightarrow \Re$ is any real-valued function, which can be discontinuous, non-differentiable; $\vec{X} = [x_1, x_2, x_3, \ldots, x_D]$ is the solution vector of the problem to be optimized; and D is the dimension of the problem. The objective of the minimization problem is to find an optimal solution vector $\vec{X}^*$, which ensure that for all $\vec{X} \in \Omega$, there is $f(\vec{X}^* < f(\vec{X}))$.

2.2. Differential Evolution Algorithm

Similar with standard Evolutionary Algorithms (EA), DE also includes three operations: mutation, crossover and selection. Unlike standard evolutionary algorithms, the DE algorithm mainly uses differential mutation operations to achieve disturbance to the population, while EA relies more on the generation of offspring randomly to disturb the population.

The classic DE algorithm flow is as follows:

- Initial population: NP solution vectors are generated randomly in solution space, of which the ith solution vector $\vec{X}_{i,G}$ express as:

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \ldots, x_{D,i,G}], \qquad (2)$$

where $i = 1, 2, \ldots, NP$ is the index of solution vector, D is the dimension of the problem to be optimized, and G is the current evolutionary iteration, which the initial value is $G = 0$.

- Mutation: there are 5 commonly used mutation strategies in DE, of which the most widely used is the strategy *DE/rand/1* which generate the mutation individual according to the following method:

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}), \qquad (3)$$

where $\vec{V}_{i,G}$ is the mutation vector generated by $\vec{X}_{i,G}$, r_1^i, r_2^i and r_3^i are mutually exclusive indices randomly chosen from solution space, which are also different from i, and F is the scaling factor which control the magnitude of the difference component.

- Crossover: to enhance the potential diversity of the population, a crossover operation comes into play after mutation. There 2 crossover strategies in DE, *exponential* and *binomial*, of which the binomial can be expressed as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if}(rand_{i,j} \leq Cr \text{ or } j = j_{rand}), \\ x_{j,i,G}, & \text{otherwise,} \end{cases} \qquad (4)$$

where $u_{j,i,G} \in \vec{U}_{i,G}$ is the element of new solution vector generated by mutation, $rand_{i,j}$ is a uniformly distributed random number in interval $[0,1]$, j_{rand} is a randomly chosen index, which ensures the it can get at least one element from $\vec{V}_{i,G}$, and Cr is cross factor which control the probability of crossover.

- Selection: the greedy selection mechanism is used to determine the solution vectors for the next iteration, the method of selection is as follows:

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G}, & \text{if } f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}), \\ \vec{X}_{i,G}, & \text{if } f(\vec{U}_{i,G}) > f(\vec{X}_{i,G}). \end{cases} \qquad (5)$$

The DE algorithm searches the solution space by mutation operation, realizes the information exchange of the solution vector by crossover operation, and obtains the better solution vector through the selection operation. Through the iterative mechanism, DE algorithm can complete the directional search of the solution space. With the increase of the size of the optimization problem, especially the increase of the dimension of the problem, the DE algorithm is easy to fall into the local optimal, and leads to premature convergence.

3. State Evaluation Adaptive Differential Evolution Algorithm

Aim at the disadvantage of DE in high-dimensional optimization problem, of which being easy to fall into local optimal and premature convergence, a State Evaluation Adaptive Differential Evolution Algorithm (SEADE) is proposed based on DE algorithm, with introducing external archiving mechanism and opposition-based population, and through state evaluation which include population distribution based state evaluation and sliding windows based evolutionary state evaluation.

3.1. External Archiving Mechanism and Opposition-based Population

The external archiving mechanism provides information guidance for subsequent evolution by stores the relevant information after evolutionary iteration. The information External archived includes the current optimal value and the optimal value of each generation of the nearest N generation, where $N = L + 2$, and L is the length of sliding window. In order to make the algorithm jump out of the local optimal, the SEADE algorithm introduces the opposition-based population [20] and [21].

Definition 1. *Let x be a real number defined in the closed interval $[a, b]$, i.e., $x \in [a, b]$. Then the opposite number $\breve{x}$ of x may be defined as:*

$$\breve{x} = a + b - x. \qquad (6)$$

Definition 2. *Opposition-based population OX of population X is defined as:*

$$OX_{j,i} = a_{j,i} + b_{j,i} - X_{j,i}, \qquad (7)$$

where $i = 1, 2, \ldots, NP$ is the index of solution vector of population, NP is the number of solution vectors, $j = 1, 2, \ldots, D$ is the dimension index of each solution vector, D is the dimension of the problem to be optimized, $a_{j,i}$ and $b_{j,i}$ indicate the upper boundary and lower boundary of dimension j of the ith solution vector, respectively. The opposition-based population can greatly increase the diversity of the population by complementary movement of the population, which has excellent ability to jump out of the local optimal.

3.2. Population-Based Distribution State Evaluation

The distribution of the population is very important for guiding the algorithm when optimizing the specific problem. Through distribution state evaluation on every dimension of the population, SEADE can get real-time search status, and provide appropriate control parameters for the sequent evolution iteration.

Definition 3. *The population center is the mean of each solution vector of the current population in each dimension, i.e., the population center of the Gth iteration is defined as $\vec{X}_G^M$:*

$$\vec{X}_G^M = \frac{1}{NP} \sum_{i=1}^{NP} x_{j,i,G}, \qquad (8)$$

where j is the dimension of the problem to be optimized, and NP is the total number of all solution vectors in the current iteration.

Definition 4. *The deviation from the population center of solution vector is defined as the distance between the solution vectors and population center in each dimension, which can express as:*

$$\vec{S}_{i,G} = \frac{\vec{X}_{i,G} - \vec{X}_G^M}{\vec{X}_{\max} - \vec{X}_{\min}}, \qquad (9)$$

where $\vec{X}_{\max}$ and $\vec{X}_{\min}$ is the upper boundary and lower boundary of the solution vector in each dimension.

Definition 5. *The population distribution state DS_G of the Gth iteration express as:*

$$DS_G = \frac{1}{NP} \sum_{i=1}^{NP} \left(\frac{1}{D} \sum_{j=1}^{D} \vec{S}_{i,G} \right), \qquad (10)$$

where NP is the total number of all solution vectors in the current iteration, and D is the dimension of the problem to be optimized. For $\vec{S}_{i,G}$ is a normalized vector according to Eq. (9), the population distribution state calculated by Eq. (10) is in the closed interval $[0, 1]$, i.e., $DS_G \in [0, 1]$.

Definition 6. *The deviation from the optimal is defined as the distance between the optimal solution vector and population center in each dimension, which can express as:*

$$\vec{S}_{best} = \frac{\vec{X}_{best} - \vec{X}_G^M}{\vec{X}_{\max} - \vec{X}_{\min}}, \qquad (11)$$

where $\vec{X}_{best}$ is the optimal solution vector stored in external archive so far.

The deviation from the population center of solution vector $\vec{S}_{i,G}$ reflects the search distribution between the

current solution vector and other solution vectors during the proceeding of evolution. The deviation from the optimal $\vec{S}_{best}$ reflects the search distribution between the current population and the optimal solution vector. The population distribution state DS_G of the Gth iteration evaluates the search status of population from a unitary perspective, which indicates that the population distribution is relatively concentrated when DS_G tends to 0, and the population distribution is relatively dispersive when DS_G tends to 1.

3.3. Sliding Window Based Evolutionary State Evaluation

In order to evaluate the evolutionary state, and judge whether the algorithm into a local optimal or not, SEADE algorithm introduces the method of sliding window. By combining with the external archive, an evaluation mechanism based on sliding window is established in SEADE.

Definition 7. *The evolutionary state window is the mean of the best archive within the window, i.e., the evolutionary state window W_G of the Gth iteration is defined as:*

$$W_G = \frac{1}{L} \sum_{n=0}^{L-1} f(\vec{X}_{Gbest,G-n}), \qquad (12)$$

where L is the length of window, $\vec{X}_{Gbest,G}$ is the global optimal individual in external archive in the Gth iteration, and $f(\cdot)$ is the fitness function.

Definition 8. *The coefficient of evolutionary state PS_G of the Gth iteration express as:*

$$W_G = \frac{W_G - W_{G-1}}{W_{G-1} - W_{G-2} + \epsilon}, \qquad (13)$$

where ϵ a small number, which is to avoid the denominator in the Eq. (13) is 0.

Its indicates in Eq. (13) that algorithm is in stagnation state when $PS_G = 0$, and in slow search state when $0 < PS_G < 1$, and in accelerated search state when $PS_G > 1$.

3.4. SEADE Algorithm

The SEADE algorithm adjusts the control parameters according to the evaluation of the distribution and evolution state of the population. The two threshold DS_G^T and PS_G^T are introduced to control the parameters, which correspond to the population distribution and evolutionary state respectively. The control parameters mainly point at differential scaling factor F and

crossover factor Cr, and the control strategies including the following:

$$F_i^{(G+1)} = (1 - |\vec{S}_{i,G} - \vec{S}_{best}|) \cdot \text{rand}(0,1), \qquad (14)$$

$$Cr_i^{(G+1)} = |\vec{S}_{i,G} - \vec{S}_{best}| \cdot \text{rand}(0,1), \qquad (15)$$

$$F_i^{(G+1)} = (1 - |\vec{S}_{i,G}|) \cdot \text{rand}(0,1), \qquad (16)$$

$$F_i^{(G+1)} = |\vec{S}_{i,G}| \cdot \text{rand}(0,1), \qquad (17)$$

where F_i^{G+1} and Cr_i^{G+1} is the differential scaling factor and crossover factor of the ith solution vector in $(G+1)$th iteration respectively, $\vec{S}_{i,G}$ and $\vec{S}_{best}$ is the deviation from population center and the optimal solution vector in Gth iteration respectively, and $\text{rand}(0,1)$ is a uniformly distributed random number in interval $(0,1)$.

In the condition of $PS_G = 0$, randomly initialize the population and go to the next iteration when $DS_G < DS_G^T$, and generate the opposition-based population and go to the next iteration when $DS_G \geq DS_G^T$.

In the condition of $0 < PS_G < PS_G^T$, update the differential scaling factor and crossover factor using Eq. (14) and Eq. (15) when $DS_G < DS_G^T$, and using Eq. (16) and Eq. (17) when $DS_G \geq DS_G^T$.

In the condition of $0 < PS_G \geq PS_G^T$, update the differential scaling factor and crossover factor using Eq. (14) and Eq. (17) when $DS_G < DS_G^T$, and using Eq. (16) and Eq. (15) when $DS_G \geq DS_G^T$.

The threshold DS_G^T and PS_G^T is adjusted according to the evolution proceeding. The strategies including the follows:

$$DS_{G+1}^T = 0.8 \cdot DS_G^T, \qquad (18)$$

$$DS_{G+1}^T = 0.2 + 0.8 \cdot DS_G^T, \qquad (19)$$

$$PS_{G+1}^T = 0.8 \cdot PS_G^T, \qquad (20)$$

$$PS_{G+1}^T = 0.2 + 0.8 \cdot PS_G^T, \qquad (21)$$

where DS_{G+1}^T and PS_{G+1}^T is the threshold after adjustment.

The DS_G^T is adjusted according to Eq. (19) after continuous N times stochastic initialization of population, and according to Eq. (18) after continuous N times generate opposition-based population. The PS_G^T is adjusted according to Eq. (21) after PS_G continuous N times fall in interval $(0, PS_G^T)$, and according to Eq. (20) after PS_G continuous N times fall in interval $(PS_G^T, +\infty)$.

The basic flow of the SEADE algorithm is shown in Tab. 1.

Tab. 1: SEADE Algorithm Flowchart.

SEADE Algorithm Flowchart
Step 1. Initializing population and related parameters
Step 2. Compute the fitness value of each initial vector, save the initial optimal value and vector to external archive
Step 3. The iteration is performed when the termination condition is not satisfied

Step 3.1 Differential mutation operation
    Step 3.2 Crossover operation
    Step 3.3 Selection operation
    Step 3.4 Evaluate the population, store the evolutionary status information and update the current optimal
    Step 3.5 Control parameters are updated using population distribution state and evolutionary state

Step 4. Output optimal value and corresponding vector

3.5. Experiments

In order to verify the performance and effectiveness of the SEADE algorithm, simulation comparison experiments is carried out. The test functions used in the experiments are listed in appendix. The contrast algorithms used in the experiments are standard DE, jDE and ODE. The experiments were carried out 50 times in each group to ensure the reliability of the solution.

The parameters for each algorithm are set as follows: the number of solution vectors of the population is 10 times of the dimension of the optimization problem, i.e., the number of solution vectors of population is 300 when the dimension of the optimization problem is 30. The number of iterations is 1000, and not any other termination conditions were set. The scaling factor F of DE is a random number in interval $(0.5, 1)$, and the crossover factor Cr is a random number in interval $(0.8, 1)$. The mutation and crossover of DE adopt the strategy of $DE/best/1/exp$. jDE and ODE adopt the recommended parameters given by literature [8] and [14] respectively. Table 2 is the experimental result, in which Mean represents the mean value of the optimal results obtained by the 50 repeated tests, and $StdDev$ is the corresponding standard deviation.

The experimental results in Tab. 2 show that the SEADE algorithm is better than the other three algorithms in the optimization effect, of which the average values indicate that the SEADE algorithm has better performance, and the standard deviation shows that the SEADE algorithm is more stable in the process of optimization. Figure 1, Fig. 2, Fig. 3 and Fig. 4 shows the contrast of average evolution process of the four algorithms with 50 repeated experiments on the test functions f_1, f_3, f_4, and f_9. The SEADE algorithm has faster convergence speed relative to the other three algorithms from the diagrams.
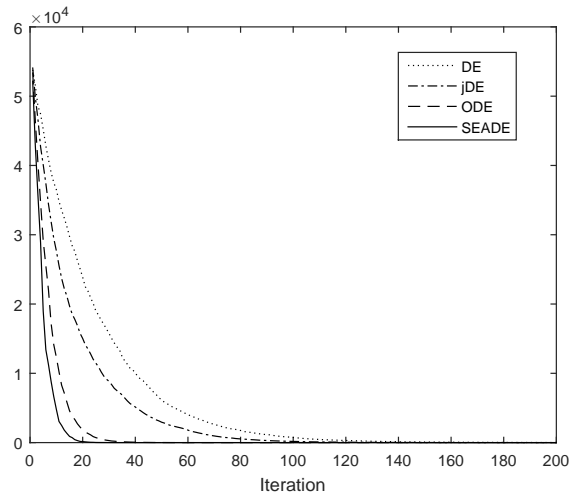

Fig. 1: Contrast of average evolution process on test function f_1 with 50 repeated experiments.
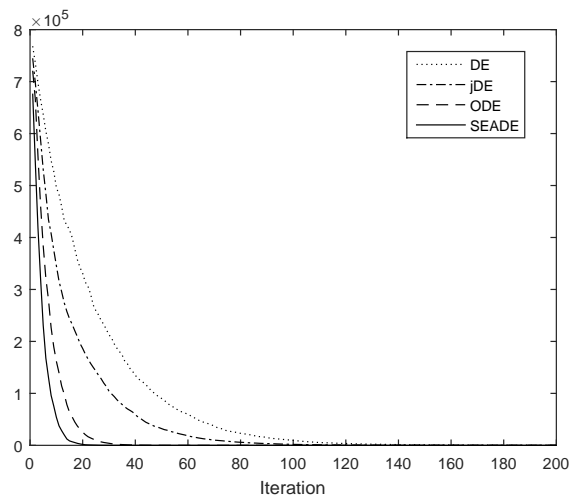

Fig. 2: Contrast of average evolution process on test function f_3 with 50 repeated experiments.
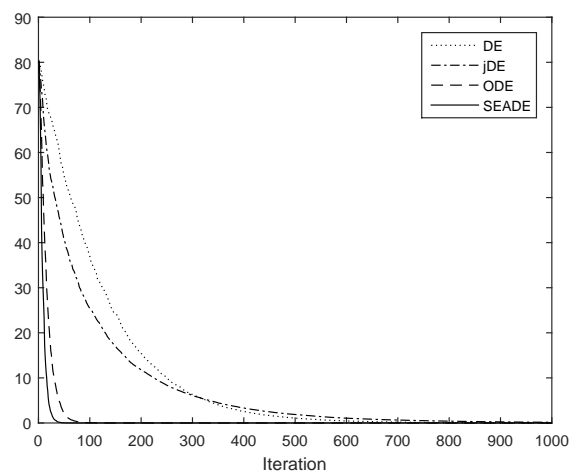

Fig. 3: Contrast of average evolution process on test function f_4 with 50 repeated experiments.

Tab. 2: The contrast of SEADE and other 3 algorithms (50 times repeated experiments).

|	DE		jDE		ODE		SEADE	
	Mean	*StdDev*	*Mean*	*StdDev*	*Mean*	*StdDev*	*Mean*	*StdDev*
f_1	$1.5259e-14$	$2.9492e-14$	$4.7680e-21$	$5.8081e-21$	$7.0837e-94$	$3.5687e-93$	$1.2458e-190$	0
f_2	$1.7774e-88$	$6.3323e-88$	$4.8831e-27$	$2.2260e-26$	$1.7109e-203$	0	$6.5074e-246$	0
f_3	$1.3992e-13$	$2.6509e-13$	$5.7435e-20$	$1.0744e-19$	$2.6812e-92$	$1.5814e-91$	$1.0216e-203$	0
f_4	0.0130	0.0080	0.1526	0.1620	$2.1560e-32$	$1.2318e-31$	$3.8576e-90$	$2.6280e-89$
f_5	0.4204	1.2054	15.3271	14.7124	0.0015	0.0021	$1.0821e-04$	$2.7317e-03$
f_6	0	0	0	0	0	0	0	0
f_7	0.0618	0.0140	0.0202	0.0054	0.0095	0.0041	$2.8861e-04$	$2.1396e-03$
f_8	$-5.7867e+169$	-	$-inf$	-	$-3.7067e+89$	$1.4335e+90$	$-3.5025e+105$	$1.6819e+106$
f_9	21.3923	2.7907	0.1194	0.3835	0	0	0	0
f_{10}	$3.4666e-08$	$3.1107e-08$	$1.3244e-11$	$8.8224e-12$	$4.4409e-15$	$1.0151e-15$	$8.8818e-16$	$1.5209e-15$
f_{11}	$6.6169e-16$	$1.7712e-15$	0.0024	0.0172	0	0	0	0
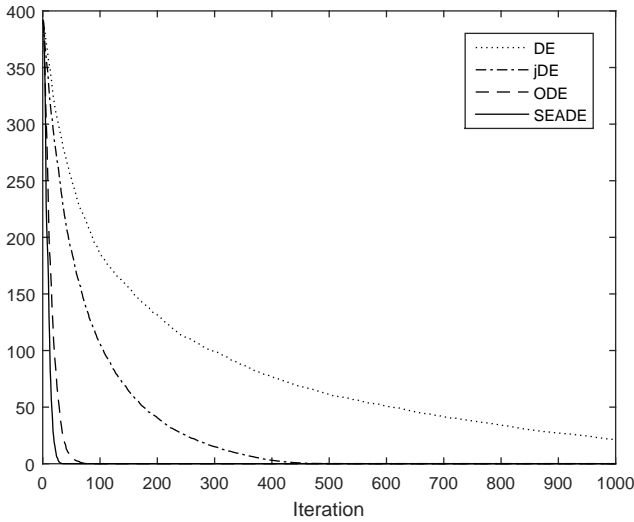
Fig. 4: Contrast of average evolution process on test function f_9 with 50 repeated experiments.

4. Optimum Design of FIR Digital Filter Using SEADE

For the N order FIR digital filter, the transfer function can be expressed as:

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n}, \qquad (22)$$

where $h(0), h(1), \ldots, h(N-1)$ is the coefficient of FIR digital filter. Take $z = e^{j\omega}$, the frequency response of the filter is:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n)e^{-j\omega n}. \qquad (23)$$

If the ideal frequency response of the FIR digital filter is recorded as $|H_d(e^{j\omega})|$, then at the discrete points $\{\omega_i | i = 1, 2, \ldots, M\}$, the sum of square of the error of the amplitude $|H(e^{j\omega})|$ of designed filter and the amplitude $|H_d(e^{j\omega})|$ of of ideal filter is:

$$E = \sum_{i=1}^{M} (|H(e^{j\omega_i})| - |H_d(e^{j\omega_i})|)^2. \qquad (24)$$

According to Eq. (23) and Eq. (24), it can conclude that

$$E = \sum_{i=1}^{M} \left[\left| \sum_{n=0}^{N-1} h(n)e^{-j\omega_i n} \right| - \left| H_d(e^{j\omega_i}) \right| \right]^2. \qquad (25)$$

Thus the optimum design of FIR digital filter can be translated into the minimization problem of the sum of square of the error between the designed frequency response and ideal frequency response.

5. Simulation

In order to verify the effectiveness of the proposed SEADE algorithm in the optimization design of FIR digital filters, the simulation experiments of FIR digital filter design are carried out by using MATLAB on computer. The Parks-McClellan algorithm [6], PSO algorithm [7] and standard DE algorithm are used as the contrast algorithms to SEADE algorithm in the experiments. The size of population in the PSO algorithm is set to 10 times of the order of the filter, and the coefficient of inertia weight $w = 0.9$, and linearly damped to $w = 0.4$ by iteration process. The acceleration constant of PSO is set to 2, i.e., $c_1 = c_2 = 2$. The size of population of DE is set to 10 times of the order of the filter, the scaling factor F is set to a random number in interval $(0.5, 1)$, and the crossover factor Cr is set to a random number in interval $(0.8, 1)$.

5.1. Optimal Design of Highpass FIR Digital Filter

For the design of 30 order high pass FIR digital filter, which the technical index is

$$H_d = \begin{cases} 1, & 0.52\pi \leq \omega \leq \pi, \\ 0, & 0 \leq \omega \leq 0.48\pi. \end{cases} \qquad (26)$$

The sampling of ω in interval $[0, \pi]$, and the total sampling points are $N = 64$, i.e.,

$$\omega_i = \frac{\pi \cdot i}{N-1}, i = 0, 1, 2, \ldots, N-1. \quad (27)$$

Then the amplitude frequency sampling point of the corresponding high pass FIR filter is

$$H_d(i) = \left[\underbrace{0, 0, \ldots, 0}_{31}, 0.3016, 0.6984, \underbrace{1, 1, \ldots, 1}_{31} \right]. \quad (28)$$

The result of contrast experiment is shown in Fig. 5 and Fig. 6, and the results of 50 times contrast experiments between DE, PSO and SEADE are listed in Tab. 3.
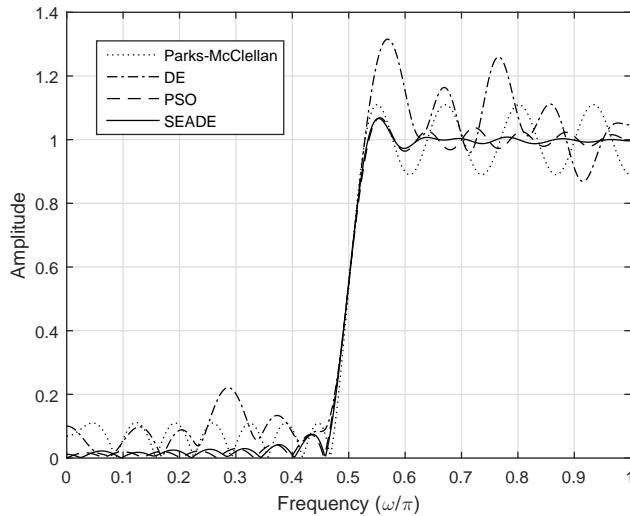

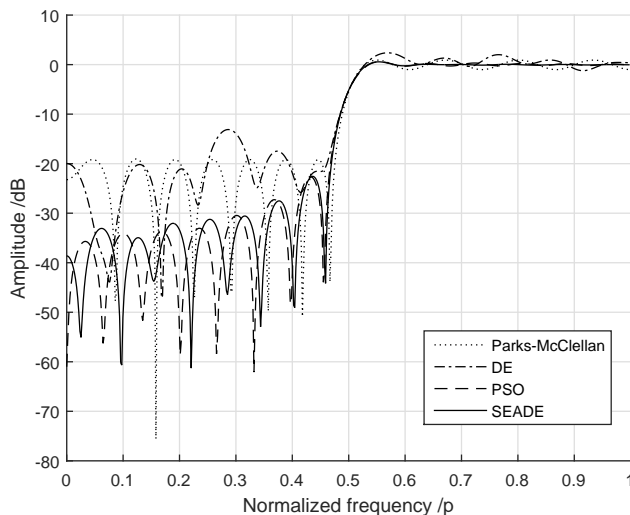**Fig. 5:** Contrast of amplitude frequency response of 30 order high pass FIR digital filter.

Fig. 6: Contrast of log amplitude frequency response of 30 order high pass FIR digital filter.

The figures and table show that the SEADE can get more optimized parameters to design the FIR filter than the methods of Parks-McClellan, standard DE and PSO.

Tab. 3: 50 times repeated contrast experiments.

	Optimal	Mean	Standard deviation
DE	0.9235	1.0032	0.0725
PSO	0.0947	0.1022	0.0091
SEADE	0.0848	0.0893	0.0057

5.2. Optimal Design of Bandpass FIR Digital Filter

For the design of 30 order bandpass FIR digital filter, which the technical index is

$$H_d(e^{j\omega}) = \begin{cases} 0, & 0 \le \omega \le 0.28\pi, 0.72\pi \le \omega\pi \\ 1, & 0.32\pi \le \omega \le 0.68\pi \end{cases}. \quad (29)$$

The sampling of ω in interval $[0, \pi]$, and the total sampling points are $N = 64$, i.e.,

$$\omega_i = \frac{\pi \cdot i}{N-1}, i = 0, 1, 2, \ldots, N-1. \quad (30)$$

Then the amplitude frequency sampling point of the corresponding bandpass FIR filter is

$$H_d(i) = \left[\underbrace{0, 0, \ldots, 0}_{18}, 0.1429, 0.5397, 0.9365, \right.$$
$$\left. \underbrace{1, 1, \ldots, 1}_{22}, 0.9365, 0.5397, 0.1429, \underbrace{0, 0, \ldots, 0}_{18} \right]. \quad (31)$$

The result of contrast experiment is shown in Fig. 7 and Fig. 8, and the results of 50 times contrast experiments between DE, PSO and SEADE are listed in Tab. 4.
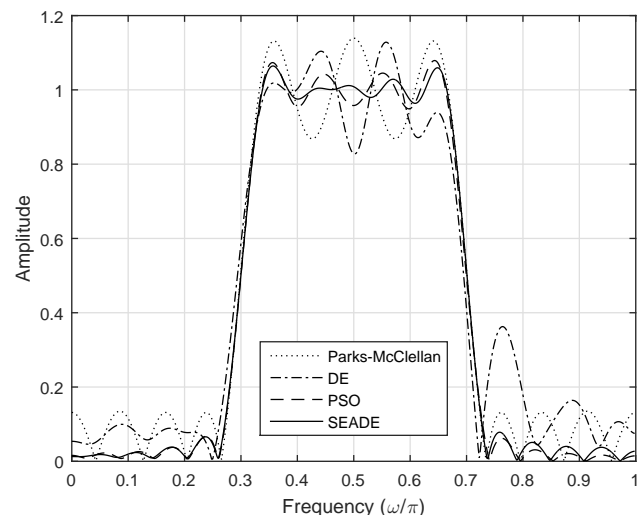

Fig. 7: Contrast of amplitude frequency response of 30 order bandpass FIR digital filter.

From the simulation results, the SEADE algorithm is better than the 3 other methods including Parks-McClellan, PSO and DE algorithm in the optimum
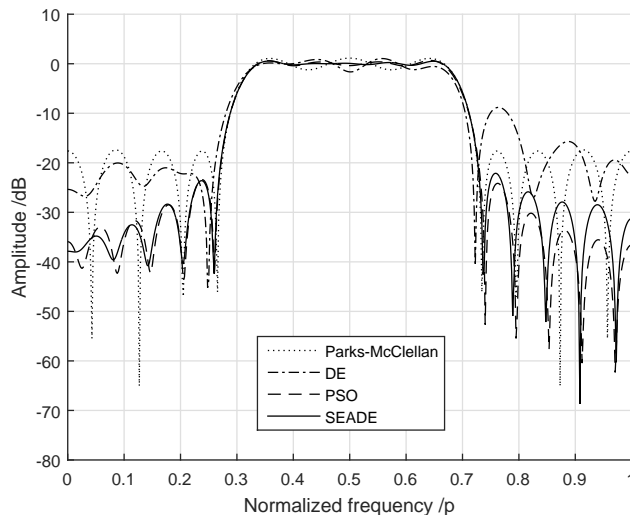
Fig. 8: Contrast of log amplitude frequency response of 30 order bandpass FIR digital filter.

Tab. 4: 50 times repeated contrast experiments.

	Optimal	Mean	Standard deviation
DE	0.9680	1.0135	0.0536
PSO	0.1597	0.1762	0.0166
SEADE	0.1326	0.1491	0.0084

design of 30 order high pass and bandpass FIR digital filter. The amplitude frequency response curve show that the optimization parameters of filter with SEADE algorithm is the best of 4 methods, and the 50 times of repeated tests show that the SEADE algorithm is much better than PSO and DE in the optimal value, average value and standard deviation.

In conclusion, it can achieve satisfactory performance with SEADE in the optimized design of FIR digital filter, and the experiments show that the optimization method is an effective design method of FIR digital filter.

6. Conclusion

As a very effective computational intelligence optimization algorithm, DE algorithm provides a new idea for the optimization of many nonlinear, non-differentiable and multi peak complex problems. Aim at standard DE algorithm is easy to fall into local optimal and premature convergence in solving high-dimensional optimization problems, we proposed a method based on state evaluation adaptive differential evolution algorithm SEADE, the simulation experiments show that the SEADE algorithm has been significantly improved in accuracy and stability in solving optimization problems. And in the optimum design of FIR digital filter, the SEADE algorithm is also proved very effective and has good performance.

References

[1] RABINER, L. R., J. H. MCCLELLAN and T. W. PARKS. FIR digital filter design techniques using weighted Chebyshev approximation. *Proceedings of the IEEE.* 1975, vol. 63, iss. 4, pp. 595–610. ISSN 0018-9219. DOI: 10.1109/PROC.1975.9794.

[2] YONG, L. and S. PARKER. Discrete coefficient FIR digital filter design based upon an LMS criteria. *IEEE Transactions on Circuits and Systems.* 1983, vol. 30, iss. 10, pp. 723–739. ISSN 0098-4094. DOI: 10.1109/TCS.1983.1085295.

[3] DEMBO, A. and D. MALAH. Generalization of the window method for FIR digital filter design. *IEEE Transactions on Acoustics Speech and Signal Processing.* 1984, vol. 32, iss. 5, pp. 1081–1083. ISSN 0096-3518. DOI: 10.1109/TASSP.1984.1164431.

[4] SUMMERS B., G. D. CAIN and A. YARDIM. FIR digital filter design using non-equispaced frequency sampling. *Electronics Letters.* 1989, vol. 25, iss. 5, pp. 338–339. ISSN 0013-5194. DOI: 10.1049/el:19890235.

[5] CHI, C. Y. and S. L. CHIOU. A new WLS Chebyshev approximation method for the design of FIR digital filters with arbitrary complex frequency response. *Signal Processing.* 1992, vol. 29, iss. 3, pp. 335–347. ISSN 0165-1684. DOI: 10.1016/0165-1684(92)90091-A.

[6] SELESNICK, I. W. and C. S. BURRUS. Exchange algorithms that complement the Parks-McClellan algorithm for linear-phase FIR filter design. *IEEE Transactions on Circuits and Systems II Analog and Digital Signal Processing.* 1997, vol. 44, iss. 2, pp. 137–143. ISSN 1057-7130. DOI: 10.1109/82.554455.

[7] HUI, L., Z. AN, Z. MIN and X. QI. Particle Swarm Optimization Algorithm for FIR Digital Filters Design. *ACTA Electronica SINICA.* 2005, vol. 33, iss. 7, pp. 1338–1341. ISSN 0372-2112.

[8] NGAMTAWEE, R. and P. WARDKEIN. Linear-phase FIR filter design using PSO method with Zero-phase Pre-design. In: *International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON).* Krabi: IEEE, 2013, pp. 1–5. ISBN 978-1-4799-0546-1. DOI: 10.1109/ECTI-Con.2013.6559652.

[9] STORN, R. and K. PRICE. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization.* 1997, vol. 11, iss. 4, pp. 341–359. ISSN 0925-5001. DOI: 10.1023/A:1008202821328.

[10] DAS, S. and P. N. SUGANTHAN. Differential evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation.* 2011, vol. 15, iss. 1, pp. 4–31. ISSN 1089-778X. DOI: 10.1109/TEVC.2010.2059031.

[11] EIBEN, A. E., R. HINTERDING and Z. MICHALEWICZ. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation.* 1999, vol. 3, iss. 2, pp. 124–141. ISSN 1089-778X. DOI: 10.1109/4235.771166.

[12] KARAFOTIAS, G., M. HOOGENDOORN and A. E. EIBEN. Parameter Control in Evolutionary Algorithms: Trends and Challenges. *IEEE Transactions on Evolutionary Computation.* 2015, vol. 19, iss. 2, pp. 167–187. ISSN 1089-778X. DOI: 10.1109/TEVC.2014.2308294.

[13] CHANG, C. S. and D. Y. XU. Differential evolution based tuning of fuzzy automatic train operation for mass rapid transit system. *IEE Proceedings - Electric Power Applications.* 2000, vol. 147, iss. 3, pp. 206–212. ISSN 1350-2352. DOI: 10.1049/ip-epa:20000329.

[14] JUNHONG, L. and J. LAMPINEN. A fuzzy adaptive differential evolution algorithm. *Soft Computing: A Fusion of Foundations, Methodologies and Applications.* 2005, vol. 9, iss. 6, pp. 448–462. ISSN 1433-7479. DOI: 10.1109/TENCON.2002.1181348.

[15] ZHANG, J. and A. C. SANDERSON. JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation.* 2009, vol. 13, iss. 5, pp. 945–958. ISSN 1089-778X. DOI: 10.1109/TEVC.2009.2014613.

[16] SHIH, M. Y. and A. CONDE. Novel coordination of DOCRs using trigonometric differential evolution algorithm. *IEEE Latin America Transactions.* 2015, vol. 13, iss. 5, pp. 1605–1611. ISSN 1548-0992. DOI: 10.1109/TLA.2015.7112021.

[17] BREST, J., S. GREINER, B. BOVSKOVIC, M. MERNIK and V. ZUMER. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation.* 2006, vol. 10, iss. 6, pp. 646–657. ISSN 1089-778X. DOI: 10.1109/TEVC.2006.872133.

[18] QIN, A. K., V. L. HUANG and P. N. SUGANTHAN. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation.* 2009, vol. 13, iss. 2, pp. 398–417. ISSN 1089-778X. DOI: 10.1109/TEVC.2008.927706.

[19] ELSAYED, S. M., R. A. SARKER and D. L. ESSAM. An improved self-adaptive differential evolution algorithm for optimization problems. *IEEE Transactions on Industrial Informatics.* 2013, vol. 9, iss. 1, pp. 89–99. ISSN 1551-3203. DOI: 10.1109/TII.2012.2198658.

[20] RAHNAMAYAN, S., H. R. TIZHOOSH and M. M. A. SALAMA. Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation.* 2008, vol. 12, iss. 1, pp. 64–79. ISSN 1089-778X. DOI: 10.1109/TEVC.2007.894200.

[21] WANG, W., H. WANG, H. SUN and S. RAHNAMAYAN. Using opposition-based learning to enhance differential evolution: A comparative study. In: *2016 IEEE Congress on Evolutionary Computation (CEC).* Vancouver: IEEE, 2016, pp. 71–77. ISBN 978-1-5090-0623-6. DOI: 10.1109/CEC.2016.7743780.

About Authors

Yong WANG was born in Huaihua, China. He received his M.Sc. from Hunan University in 2009. His research interests include intelligent information processing and information fusion.

Zhaosheng TENG was born in Huaihua, China. He received his Ph.D. from Hunan University in 1998. His research interests include power quality measurement and information fusion.

He WEN was born in Yiyang, China. He received his Ph.D. from Hunan University in 2009. His research interests include information fusion and electrical measurement.

Jianmin LI was born in Ganzhou, China. He received his M.Sc. from Hunan University in 2012. His research interests include power quality measurement and intelligent information processing.

Radek MARTINEK is currently an Associate Professor with the Department of Cybernetics and Biomedical Engineering, Electrical Engineering and Computer Science, VSB–Technical University of Ostrava.

Appendix A
Test Functions

- $$f_1(x) = \sum_{i=1}^{n} x_i^2,$$

  dimension $n = 30$, variable $x_i \in [-100, 100]^n$ and the optimal is 0.

- $$f_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|,$$

  dimension $n = 3$, variable $x_i \in [-10, 10]^n$ and the optimal is 0.

- $$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j^2 \right),$$

  dimension $n = 30$, variable $x_i \in [-100, 100]^n$ and the optimal is 0.

- $$f_4(x) = \max\{|x_i|, 1 \le i \le D\},$$

  dimension $n = 30$, variable $x_i \in [-100, 100]^n$ and the optimal is 0.

- $$f_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right],$$

  dimension $n = 30$, variable $x_i \in [-30, 30]^n$ and the optimal is 0.

- $$f_6(x) = \sum_{i=1}^{n} \left(\lfloor x_i + 0.5 \rfloor \right)^2,$$

  dimension $n = 30$, variable $x_i \in [-100, 100]^n$ and the optimal is 0.

- $$f_7(x) = \sum_{i=1}^{n} i x_i^4 + \mathrm{random}(0, 1),$$

  dimension $n = 30$, variable $x_i \in [-100, 100]^n$ and the optimal is 0.

- $$f_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}),$$

  dimension $n = 30$, variable $x_i \in [-500, 500]^n$ and the optimal is -12569.

- $$f_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10 \cos(2\pi x_i) + 10 \right],$$

  dimension $n = 30$, variable $x_i \in [-5.12, 5.12]^n$ and the optimal is 0.

- $$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\left(1/n \sum_{i=1}^{n} x_i^2 \right)} \right)$$
  $$- \exp\left((1/n) \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + \exp(1),$$

  dimension $n = 30$, variable $x_i \in [-32, 32]^n$ and the optimal is 0.

- $$f_{11}(x) = (1/400) \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1,$$

  dimension $n = 30$, variable $x_i \in [-100, 100]^n$ and the optimal is 0