# Design of 2-D Recursive Filters Using Self-adaptive Mutation Differential Evolution Algorithm

**Lianghong Wu** [1,2] **, Yaonan Wang** [2] **, Xiaofang Yuan** [2]

[1] *Engineering Research Center of Advanced Mining Equipment, Ministry of Education, Hunan University of Science and Technology,*
*Xiangtan, Hunan 411201, China*
*E-mail: lhwu@hnust.cn*

[2] *College of Electrical and Information Engineering, Hunan University,*
*Changsha, Hunan 410082, China*

## Abstract

This paper investigates a novel approach to the design of two-dimensional recursive digital filters using differential evolution (DE) algorithm. The design task is reformulated as a constrained minimization problem and is solved by an Self-adaptive Mutation DE algorithm (SAMDE), which adopts an adaptive mutation operator that combines with the advantages of the DE/rand/1/bin strategy and the DE/best/2/bin strategy. As a result, its convergence performance is improved greatly. Numerical experiment results confirm the conclusion. The proposed SAMDE approach is effectively applied to test a numerical example and is compared with previous design methods. The computational experiments show that the SAMDE approach can obtain better results than previous design methods.

*Keywords:* Two-dimensional recursive filters; Differential evolution; Self-adaptive mutation; Constrained optimization.

## 1. Introduction

Multidimensional digital signal processing is one of the important development directions of the digital signal processing. In the multidimensional signal processing, two-dimensional (recursive or non-recursive) filters are the most widely used algorithm or processor. During the last three decades, two-dimensional (2-D) digital filter design has received growing attention and has been applied in many areas such as digital image processing, seismic data processing, pattern recognition, astronomy and applied physics, robotics and mechanical engineering, etc [1], [2]. Moreover, two-dimensional filter mathematics can be used in grid methods for solving partial differential equations, distributed control, and iterative learning control, etc.

The most popular design approaches for 2-D filters are based either on appropriate transformation of 1-D filters [2-4], or on appropriate optimization techniques [5-17]. One of the major problems underlying the design task is to satisfy the stability criterion for the filter transfer function. However, most of the existing algorithms [3-10] may result in an unstable filter. Although various techniques have been proposed in order to overcome these instability problems, the outcome is likely to be a system that has a very small stability margin and therefore of no

Published by Atlantis Press

essential practical importance [5], [12], [13].

In recent years, intelligent optimization algorithms as an alternative to deal with complex and difficult real-world optimization problems are employed to solve the 2-D filter optimal design problem. V. M. Mladenov et al. used a continuous-time neural network to the design of 2-D recursive filter [11]. N. E. Mastorakis et al. proposed a new design method for two-dimensional recursive filters using genetic algorithms [12]. J. T. Tsai et al. proposed a hybrid Taguchi-genetic algorithm (HTGA) to solve the design problem of 2-D recursive digital filters and obtained better results than previous design methods [13]. S. T. Tzeng proposed an effective GA approach for designing 2-D real FIR digital filters with complex-valued frequency responses in [14]. In [15], W. Fang et al. introduced a novel quantum-behaved particle swarm optimization (QPSO) algorithm and used to solve the design problem of 2-D recursive digital filters. S. Das and A. Konar [16] investigated a novel approach to the designing of two-dimensional zero phase infinite impulse response (IIR) digital filters using a new variant of PSO called MEPSO. J. T. Tsai et al. [17] introduced a Taguchi-based immune algorithm (TBIA) and applied to solve the design problem of two-dimensional recursive digital filters.

Designing digital 2-D recursive filters is a multiparameter optimization problem, and there are numerous local optima when the number of parameters is large. Therefore, it is worthy to further apply an efficient evolutionary algorithm to solve the digital 2-D recursive filter design problems. Differential evolution (DE) algorithm, developed by Storn and Price [18,19], is a salient optimizer which can be used in a broad variety of highly nonlinear and complex optimization problems [20-21]. In 1st International Contest on Evolutionary Computation in 1997, DE has evolved in an effort to improve upon it's the third place [20], and been one of the most excellent EAs for solving the real-valued optimization problems. The DE algorithm has gradually become more popular and has been used in many practical cases such as control engineering, chemistry, electromagnetic, robot [22-27], mainly because it has demonstrated good convergence properties and is principally easy to be understood.

The contribution of this paper is to propose a novel DE algorithm to solve the 2-D recursive digital filters. In this work, the design task of 2-D recursive filters is formulated as a constrained optimization problem. The stability criterion is presented as constraints to the minimization problem. Thus, a generalized filter design framework has been illustrated, which is suitable for application of any global optimization algorithm. An self-adaptive mutation DE algorithm (SAMDE), which adopts adaptively two different mutation operator, is presented to improve the DE's convergence properties over difficult fitness landscapes. The effectiveness of the modified version is validated by using several classical Benchmark functions and a specific design problem.

The remainder of this paper is organized as follows. Section 2 provides an overview of the filter design methodology and its reformulation as a constrained minimization problem. In section 3, the DE algorithm is presented briefly at first, and then the SAMDE algorithm is introduced in detail. In section 4, numerical experiment is demonstrated to test the effectiveness of the proposed approach. The application of the proposed method to the present problem is described firstly in section 5, then, the results of applying the proposed method to a specific design problem and performance comparisons with other optimization techniques are presented. Finally, the paper is concluded in section 6.

## 2. Problem Formulation

Digital filters are broadly classified into two main categories namely, finite impulse response (FIR) filters and infinite impulse response (IIR) filters. An FIR filter is one whose impulse response is of finite duration. Generally, the output of such a filter is calculated solely from the current and previous input values. On the other hand, an IIR filter is one whose impulse response continues for ever in time. The current output of such a filter depends upon previous outputs. They are also termed as recursive filters.

Compared with FIR filters, frequency response with narrow transition band can be easily realized with IIR filters and this feature makes them suitable for a broad range of applications such as the modeling of random fields for image processing and computer vision [16]. Another potential advantage retained in IIR filters is that a given set of frequency response characteristics typically may be met by an IIR filter of considerably lower order than a corresponding FIR design [2]. Since they are more challenging to implement and allow for higher degrees of precision and practical efficiency, the 2-D IIR filter is selected as the focus of our proposed methodology. Our problem is now formulated as follows.

For design purposes, we consider the following 2-D transfer function for our filter [12-13]:

$$H(z_1,z_2)=H_0\frac{\sum_{i=0}^{L}\sum_{j=0}^{L}a_{ij}z_1^i z_2^j}{\prod_{k=1}^{K}(1+b_kz_1+c_kz_2+d_kz_1z_2)},a_{00}=1. \tag{1}$$

where $a_{ij}$, $b_k$, $c_k$ and $d_k$ are coefficients of the filter, $L$ and $K$ are orders of the numerator and denominator, respectively, with $K \geqslant L$, $H_0$ is a constant. The variables $z_1$ and $z_2$ can be interpreted as complex indeterminants in the discrete Laplace transform ($z$ transform).

Consider $M_d$, the desirable amplitude response of the 2-D filter as a function of the frequencies $\omega_1$ and $\omega_2$, $(\omega_1, \omega_2 \in [0,\pi])$. The design problem is to determine the coefficients in the numerator and denominator of Eq. (1) in such a fashion that $M(\omega_1,\omega_2)$=$H(e^{j\omega_1},e^{j\omega_2})$ follows the desired response $M_d(\omega_1,\omega_2)$ as closely as possible. Such an approximation of the desired response can be achieved by minimizing

$$\begin{aligned}J &= J(a_{ij},b_k,c_k,d_k,H_0)\\ &= \sum_{n_1=0}^{N_1}\sum_{n_2=0}^{N_2}[|M(\omega_1,\omega_2)|-M_d(\omega_1,\omega_2)]^p\end{aligned} \tag{2}$$

where $\omega_1 = (\pi n_1/N_1)$, $\omega_2 = (\pi n_2/N_2)$, and $p$ is an even positive integer (usually $p$=2 or $p$=4).

Eq.(2) can be restated as

$$J=\sum_{n_1=0}^{N_1}\sum_{n_2=0}^{N_2}\left[\left|M(\frac{\pi n_1}{N_1},\frac{\pi n_2}{N_2})\right|-M_d(\frac{\pi n_1}{N_1},\frac{\pi n_2}{N_2})\right]^p \tag{3}$$

Hence, the aim is to minimize the difference between the actual and desired amplitude response of the filter at $N_1 \times N_2$ points. For bounded input bounded output (BIBO) stability the prime requirement is that the z-plane poles of the filter transfer function should lie within the unit circle. Since the denominator contains only first-degree factors, it is known that the stability conditions are given by [1-3]

$$|b_k+c_k|-1<d_k<1-|b_k-c_k|, \quad k=1,2,\cdots,K,$$

or

$$\begin{aligned}&|b_k+c_k|-1<d_k, \quad k=1,2,\cdots,K,\\ &d_k<1-|b_k-c_k|, \quad k=1,2,\cdots,K.\end{aligned} \tag{4}$$

Thus, the design of 2-D recursive filter is equivalent to the following constrained minimization problem:

minimize

$$J=\sum_{n_1=0}^{N_1}\sum_{n_2=0}^{N_2}\left[\left|M(\frac{\pi n_1}{N_1},\frac{\pi n_2}{N_2})\right|-M_d(\frac{\pi n_1}{N_1},\frac{\pi n_2}{N_2})\right]^p \tag{5}$$

subject to the constraints

$$\begin{aligned}&|b_k+c_k|-1<d_k, \quad k=1,2,\cdots,K,\\ &d_k<1-|b_k-c_k|, \quad k=1,2,\cdots,K,\end{aligned} \tag{6}$$

where $N_1$, $N_2$, and $K$ are given positive integers, and $p$ is an even positive integer. Without loss of generally let us assume $K$=2, choose the values $p$=2, $N_1$=50, and $N_2$=50, the corresponding constrained optimization problem (5) becomes

minimize

$$J=\sum_{n_1=0}^{50}\sum_{n_2=0}^{50}\left[\left|M(\frac{\pi}{50}n_1,\frac{\pi}{50}n_2)\right|-M_d(\frac{\pi}{50}n_1,\frac{\pi}{50}n_2)\right]^2 \tag{7}$$

subjected to the constraints imposed by inequalities (6) with $k$=1, 2.

There are 15 parameters in the present problem and represented by the vector as

$X$=$(a_{01},a_{02},a_{10},a_{11},a_{12},a_{20},a_{21},a_{22},b_1,b_2,c_1,c_2,d_1,d_2,H_0)^T$

In [11], the problem has been tackled using neural networks. In [12] and [13], a binary coded GA and a hybrid GA are attempted to solve this problem. It is also been optimized by a quantum-behaved PSO and a modified PSO algorithm in [15] and [16], respectively. In this paper, a much better solution has been obtained using the improved DE algorithm.

## 3. Self-adaptive Mutation DE Algorithm

### 3.1. Basic DE Algorithm

DE is a simple yet powerful evolutionary algorithm for global optimization, which utilize $NP$ D-dimensional parameter vectors $\overline{X}^t = \{X_1^t, X_2^t, \cdots, X_{NP}^t\}$ as a population for each generation $t$, $NP$ is the population size, and $X_i^t = (x_{i1}^t, x_{i2}^t, \cdots x_{iD}^t)(i = 1, \cdots, NP)$ represents the parameter vector to be optimized. The initial population is chosen randomly and should cover the entire parameter space. At each generation, DE employs both mutation and crossover to produce one trial vector for each target vector. Then, a selection phase takes place, where each trial vector is compared to the corresponding target vector. The better one will enter the population of the next generation. For each target vector $X_i^t$, a mutant vector $V_i^t = (v_{i1}^t, v_{i2}^t, \cdots, v_{iD}^t)$ is generated as

$$V_i^t = X_{r_1}^t + F \cdot (X_{r_2}^t - X_{r_3}^t) \tag{8}$$

where $F > 0$ is a real parameter, called mutation constant or difference vector scale factor, which controls the amplification of the difference vector $(X_{r2}^t - X_{r3}^t)$ so as to avoid search stagnation. According to Storn and Price [18], the $F$ is set in (0, 2]. $r_1$, $r_2$, $r_3$ are indexes, randomly selected from the set $\{1, 2, ..., NP\}$. Note that indexes must be different from each other and from the running index $i$ so that $NP$ must be a least four.

The DE algorithm has several various strategies based on the individual to be perturbed, the number of difference vectors used in the mutation process and the type of crossover scheme. The different strategies can be denoted as DE/*x*/*y*/*z*[18,19], where

*x* specifies the individual to be perturbed. It can be "rand" or "best". "rand" means a randomly selected individual, and "best" for the individual with the best fitness from the current population.

*y* is the number of difference vectors for perturbation of *x*.

*z* stands for the type of crossover scheme, which can be "bin" or "exp". "bin" means crossover is due to independent binomial experiments, "exp" stands for crossover is performed as exponential experiments.

So the above mentioned strategy (Eq. (8)) can be denoted as DE/rand/1/bin, which is the standard or basic and widely used strategy.

Following the mutation phase, the crossover operator is applied on the population. For each mutant vector $V_i^t$, a trial vector $U_i^t = (u_{i1}^t, u_{i2}^t \cdots, u_{iD}^t)$ is generated, using the following scheme.

$$u_{ij}^t = \begin{cases} v_{ij}^t, \ rand(j) \leqslant CR \ or \ j = randn(i) \\ x_{ij}^t, \ rand(j) > CR \ and \ j \neq randn(i) \end{cases} \tag{9}$$

where $j$=1, ..., $n$. $rand(j)$ is the $j$th evaluation of a uniform random number generator within [0, 1]. $CR$ is a crossover probability constant in the range [0, 1], which has to be determined previously by the user. $randn(i) \in (1, 2, ..., n)$ is a randomly chosen index which ensures that $U_i^t$ gets at least one element from $V_i^t$. Otherwise, no new parent vector would be produced and the population would not alter.

To decide whether the trial vector $U_i^t$ should be a member of the population comprising the next generation, it is compared to the corresponding target vector $X_i^t$, and the greedy selection strategy is used in DE. The selection operator is as following.

$$X_i^{t+1} = \begin{cases} U_i^t, \ \texttt{if} \ f(U_i^t) < f(X_i^t) \\ X_i^t, \ \texttt{otherwise} \end{cases} \tag{10}$$

where $f$ is the objective function, and the minimization problem is considered here.

With the members of the next generation thus selected, the evolutionary cycle of the DE repeats until a stopping criterion is satisfied.

### 3.2. Self-adaptive Mutation Operator

Among the variants of DE, those strategies with "best" as the individual to be perturbed have fast convergence rate and good local search ability, but they will reduce the diversity of the population and lead to premature convergence. However, those strategies with "rand" as the individual to be perturbed are good for keeping diversity of the whole population and global exploration ability, but will decelerate the convergence rate seriously. Obviously, the DE/rand/1/bin strategy has good global search ability since its base vector of mutation vector is randomly selected in the population. One of highly beneficial method of DE is the strategy

DE/best/2/bin [18], which uses the best individual of the current population as the vector to be perturbed, so it has powerful local search ability and fast convergence property. A good optimization algorithm should be such one which has the global exploration ability to find as many as possible potential global optimizations at the beginning stage and the local exploitation ability at the last stage to improve precision of the solution obtained. Thus, an adaptive mutation operator combined with the advantages of strategies of DE/rand/1/bin and DE/best/2/bin is proposed in this paper. The novel mutation strategy is as the following.

$$V_i^t = \begin{cases} X_{r1}^t + F(X_{r2}^t - X_{r3}^t), \texttt{if } rand < 1-(t/T)^2 \\ X_b^t + F(X_{r1}^t - X_{r2}^t + X_{r3}^t - X_{r4}^t), \texttt{otherwise} \end{cases} \tag{11}$$

where $X_b^t$ is the individual with the best fitness from the current population, *rand* is an uniform random number generator within [0, 1], $t$ is the current evolution generation, and $T$ is the maximal generation.

The new mutation operator shows that at the beginning stage, the strategy of DE/rand/1/bin has more chance to be used, the diversity of the population and the chance to get into the global optimization are improved greatly. With the evolution going on, the probability of DE/best/2/bin strategy to be selected is increased. Consequently, the local search ability is enhanced and the accuracy is improved, the convergence rate is also speeded up. All in all, the new mutation strategy can keep the balance between the global exploration and local exploitation, and then improve the convergence rate and accuracy.

### 3.3. *Procedures of the SAMDE Algorithm*

The proposed SAMDE algorithm is summarized as follows.

**Step1**: Specify parameters of the SAMDE algorithm: population size *NP*, difference vector scale factor $F$, crossover probability constant *CR*, the maximum number of generations $T$; initialize randomly the individuals of the population and the trial vector in the given searching space. Set $t$=0.

**Step2**: Evaluate the fitness value of each individual in the current population, and find the best fitness and corresponding best individual $x_b^t$ in the current population.

**Step3**: If the stopping criteria are satisfied, then go to step 8. Otherwise, do the next steps.

**Step4**: Generate a mutant vector according to the new mutation operator (Eq.(11)) for each individual.

**Step5**: According to Eq.(9), do the crossover operation and yield a trial vector.

**Step6**: Do the selection operation in terms of Eq.(10) and generate a new population.

**Step7**: $t = t$+1, return to Step2.

**Step8**: Output the optimal solution and stop.

## 4. Numerical Experiment

To demonstrate the effectiveness of the adaptive mutation operator, a comparison among SAMDE, DE/rand/1/bin and DE/best/2/bin is carried out in this section. Four classic benchmark functions are used to estimate the performance of the proposed SAMDE algorithm. The variety of dimensions and functional forms make it possible to fairly assess the effectiveness, efficiency and accuracy of the proposed approach. The test functions are as following.

Schaffer

$$f_1 = 0.5 - \frac{\sin^2\sqrt{x_1^2+x_2^2} - 0.5}{(1+0.001(x_1^2+x_2^2))^2}, \;\; x_i \in [-100, 100] \tag{12}$$

Rosenbrock

$$f_2 = \sum_{i=1}^{30} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], \;\; x_i \in [-50, 50] \tag{13}$$

Rastrigin

$$f_3 = \sum_{i=1}^{30} [x_i^2 - 10\cos(2\pi x_i) + 10], \;\; x_i \in [-5.12, 5, 12] \tag{14}$$

Griewank

$$f_4 = \frac{1}{400}\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30}\cos(\frac{x_i}{\sqrt{i}}) + 1, \;\; x_i \in [-30, 30] \tag{15}$$

In our experiments, the following parameters are used. The mutation constant $F_r$ of DE/rand/1/bin and $F_b$ of DE/best/2/bin are the same, $F_r$=$F_b$=0.5, the crossover probability constant $CR_r$ of DE/rand/1/bin and $CR_b$ of DE/best/2/bin are 0.1 and 0.9, respectively. Our decision for using those values is based on the balance between the global exploration and local exploitation and proposed values from literature [18-21]. Also, these values were determined after performing an extensive set of experiments. The population size is 30 for $f_1$, 100 for $f_2 \sim f_4$. The maximum number of generations is 300 for $f_1$, 500 for $f_2$, 2000 for $f_3 \sim f_4$. Experiment condition is: AMD Athlon 3400+ CPU, 512M Memory, Windows XP operation system, all the program is realized in Matlab. For fair comparison, all algorithms used the same population size and maximum function evaluations for each test function. Table 1 summarizes the average results and rate of successful optimization of 50 independent runs for each test function. If the error between theory optimum and experiment result is less than $10^{-4}$, it is thought that the algorithm get the optimum.

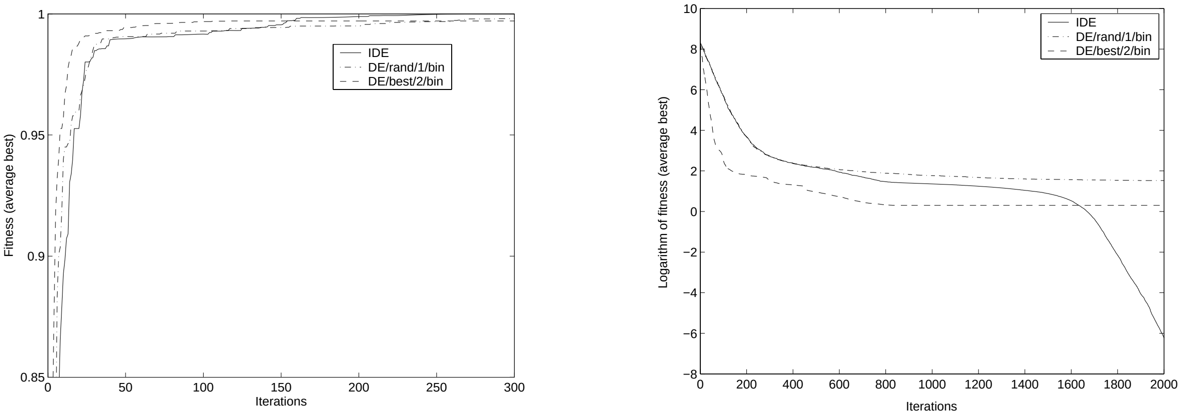
From the Table 1, it can be observed that the average best result obtained by SAMDE is more accuracy than those obtained by DE/rand/1/bin and DE/best/2/bin. In 50 independent runs, SAMDE has higher rate of successful optimization for each test function than DE/rand/1/bin and DE/best/2/bin, hence the global convergence ability of SAMDE is better than the two DE variants. Fig.1(a)$\sim$(d) are average best fitness curves of 50 independent runs. To show the difference further clearly, logarithm operation is used for the vertical axis of Fig.1(b)$\sim$(d). Observing Fig.1, we can see that at the beginning stage there is more chance to select the strategy of DE/rand/1/bin, SAMDE has similar convergence performance of DE/rand/1/bin. While with the evolution process going on, the chance to select the strategy of DE/best/2/bin is increased and the convergence speed of SAMDE is accelerated.

According to above experiment results we can claim that the new mutation strategy can keep the balance between the global exploration and local exploitation, SAMDE has obviously better convergence speed, global convergence rate and solution accuracy than DE/rand/1/bin and DE/best/2/bin.

Table 1. Average results and rate of successful optimization of 50 independent runs.

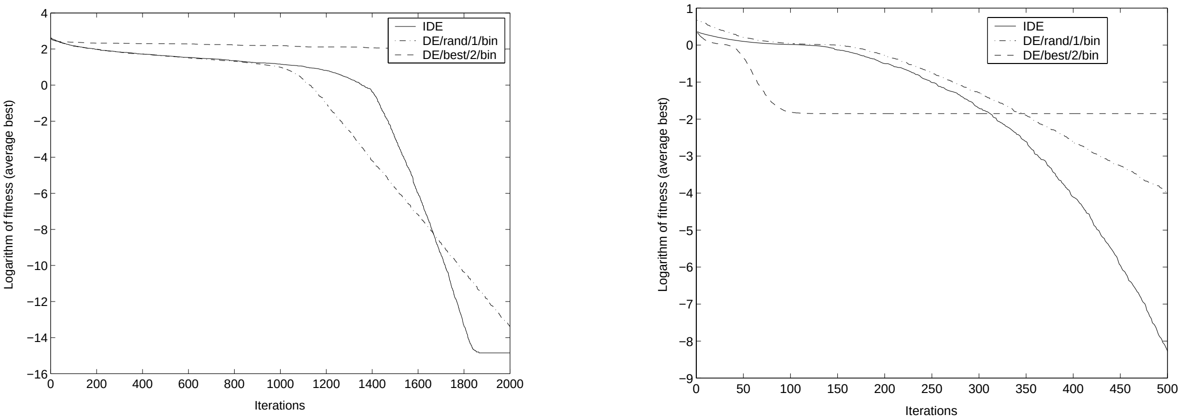
| | SAMDE | | | DE/rand/1/bin | | | DE/best/2/bin | | |
|---|---|---|---|---|---|---|---|---|---|
| function | Mean | Std | RS | Mean | Std | RS | Mean | Std | RS |
| $f_1$ | 0.999987 | 7.20e-5 | 98 | 0.998399 | 2.71e-3 | 67 | 0.993695 | 4.70e-3 | 44 |
| $f_2$ | 1.9513e-8 | 4.30e-8 | 100 | 1.0774e-4 | 4.47e-5 | 79 | 1.2548e-2 | 1.35e-2 | 35 |
| $f_3$ | 1.332e-15 | 1.60e-15 | 100 | 4.0056e-14 | 4.55e-14 | 100 | 8.9401e+1 | 6.09e+1 | 0 |
| $f_4$ | 2.8402e-8 | 4.49e-8 | 100 | 3.3172e+1 | 1.18e+1 | 0 | 1.512158 | 1.96859 | 62 |

(MB is mean best, STD is standard deviation, RS is rate of successful optimization.)



(a) $f_1$

(b) $f_2$



(c) $f_3$

(d) $f_4$

Fig. 1. Average best fitness curves of 50 runs for $f_1 \sim f_4$.

## 5. Illustrative example and comparisons

In this section, we adopt the same example to compare the performance of our presented SAMDE approach with the performance of some intelligent computation techniques such as Neural Networks (NN) [11], genetic algorithm (GA) [12], particle swarm optimization (PSO) [16] and Taguchi-based immune algorithm (TBIA) [17]. Moreover, the result of the traditional method (Lp) given in [5] is also compared with the result of SAMDE. There is often a trade-off between result and complexity. Therefore, for fair performance comparison, the same function evaluations (FEs) as the compared algorithms (except for NN and Lp) was used in the experiment.

### *5.1. Handling constraints*

The design of 2-D digital IIR filter is essentially a multi-parameter optimization problem with constraints. For optimization problems with constraints, the penalty-function approach is frequently used to handle constraints. Here a penalty-parameterless constraint-handling approach is used to handle constraints [27]. A solution $i$ is said to be preferred to a solution $j$, if any of the following conditions is true.

(a) Solution $i$ is feasible and solution $j$ is not,

(b) Solutions $i$ and $j$ are both infeasible, but solution $i$ has a smaller overall constraint violation,

(c) Solutions $i$ and $j$ are feasible and solution $i$ has better objective function value than solution $j$.

The effect of using this method is that any feasible solution has a preference than any infeasible solution. Between two feasible solutions, the one with better objective function value is preferred. However, among two infeasible solutions, the solution with a smaller constraint violation has a higher rank. This constrained-handling principle does not change the computational complexity of the SAMDE.

### *5.2. Performance metrics*

The performance of the 2-D IIR filters designed here, are evaluated in terms of the following two metrics:

(1) Pass-band magnitude ripple

$$e_{pmr} = ||H(e^{j\omega_1}, e^{j\omega_2})| - 1|, \quad \omega_1, \omega_2 \in [0, \omega_p] \quad (16)$$

(2) Stop-band attenuation

$$A(\omega_1, \omega_2) = \\ -20\log_{10}|H(e^{j\omega_1}, e^{j\omega_2})|, \quad \omega_1, \omega_2 \in [\omega_s, \pi] \quad (17)$$

where $\omega_p$ and $\omega_s$ are the pass-band and stop-band frequencies, respectively. For a good design, the maximum pass-band ripple should be low enough to ensure a distortionless production of the output signal. On the contrary, the minimum value of stop-band attenuation should be large in order to ensure the elimination of the undesired frequency components to a high degree.

### 5.3. Experiment and results

Let us consider an example of the design problem, where the user-specification for the desired circular symmetric low-pass filter response is given as

$$M_d(\omega_1,\omega_2)=\begin{cases} 1, & \sqrt{\omega_1^2+\omega_2^2}\leqslant 0.08\pi, \\ 0.5, & 0.08\pi<\sqrt{\omega_1^2+\omega_2^2}\leqslant 0.12\pi, \\ 0, & \texttt{otherwise.} \end{cases} \tag{18}$$

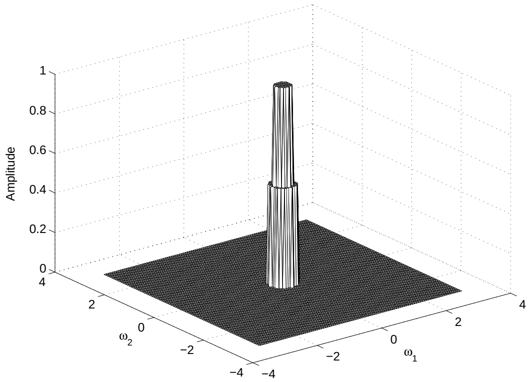


Fig. 2. Desired amplitude response of the 2-D filter.

This example refers to a low-pass filter with a small area of pass-ability. We have to give more emphasis in this small area than the other areas as in [12]. Hence, in the present design method, in order to give also more emphasis in the passable area of the filter, we introduce weight factors in Eq.(5). We use a weight 100 for $n_1$=0, $n_2$=0, e.g., for the point $\omega_1 = 0$, $\omega_2 = 0$, while the weights for other 2600 points are 1. A similar preference for the low frequencies has been made in [12]. Fig. 2 shows the desired amplitude response $M_d(\omega_1,\omega_2)$.

In our experiments, the following parameters are used. As suggested in [16], we select the initial value of the filter parameters randomly from the interval (-3, 3). In case of the proposed SAMDE algorithm, the same mutation constant and crossover probability constant are used as section 4. The population size is 50, and the maximum number of generations is 1000. That is 50,000 FEs, the same as GA [12], PSO [16] and TBIA [17] for fair performance comparison. Moreover, the difference between the actual and the desired amplitude response of the filter ($J$) is considered as the fitness function. The vector $X$ obtained through the use of the SAMDE is

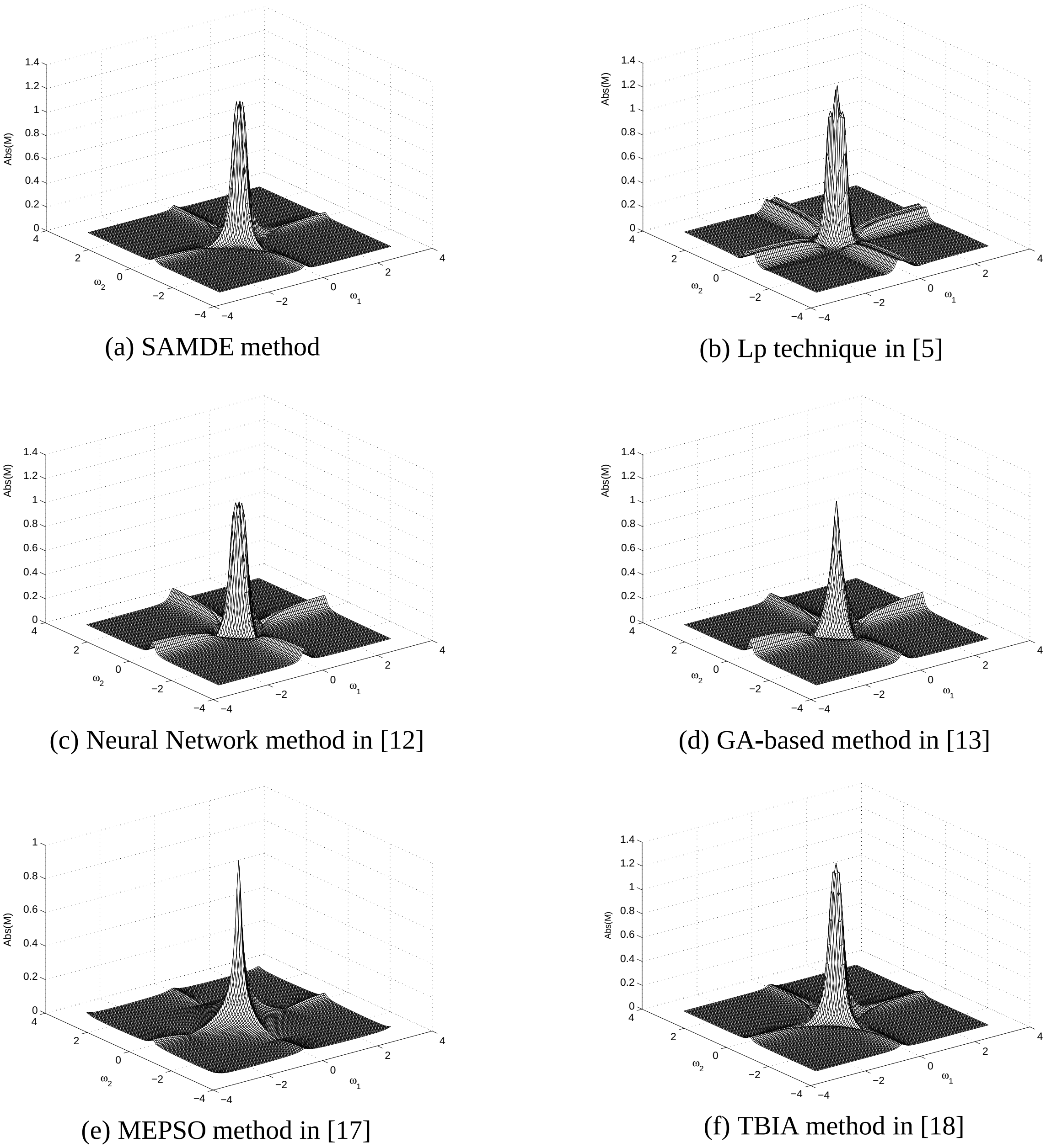
$X$=[1.83831, -1.16632, 1.44850, -2.47933, -0.036934, -0.79494, -0.42047, 2.9423, -0.89135, -0.89205, -0.89011, -0.89328, 0.80637, 0.81008, 0.00030263]$^T$.

Table 2. Performance comparison of the derived filters.

| Performance metrics | SAMDE | TBIA [17] | MEPSO [16] | GA [12] | NN [11] | Lp [5] |
|---|---|---|---|---|---|---|
| Fitness function $J$ | **2.4555** | 2.7761 | 5.7362 | 5.3083 | 3.6698 | 3.604 |
| Maximum $e_{pmr}$ | **0.1630** | 0.2976 | 0.2991 | 0.5905 | 0.2549 | 0.2708 |
| Minimum $A(\omega_1,\omega_2)$ | **29.161** | 29.158 | 19.628 | 17.434 | 19.141 | 18.953 |

Table 2 shows the performance comparison of the derived filters in terms of the performance metrics mentioned in section 5.2. From Table 2 we can see that SAMDE obtains lower value of cost function and lower value of maximum pass-band magnitude ripple than other algorithms, which means that, compared to other five algorithms, SAMDE has better convergence performance, and the filter obtained by SAMDE has minimum distortion production of output signal. Regarding to the minimum stop-band attenuation metric, SAMDE obtains the highest value, that is the filter obtained by which has better ability to eliminate the undesired frequency components. From the Table 2, it can be observed that the method MEPSO in [16] generated the largest value of cost function, while the method GA in [12] yielded the smallest value of minimum stopband attenuation and the largest value of maximum passband magnitude ripple.

The amplitude responses $|M(\omega_1,\omega_2)|$ of the considered 2D-filter using different methods are shown in Fig.3. From the amplitude responses of the designed filters in Fig.3, it can be seen that the better filter, which has desirable coefficients in the transfer function, has smaller difference between the actual and the desired amplitude responds of the filter and the flatter shape in the stopband. However, the worse filter has large error at low frequencies and the special cross shape in the stopband. A closer look at these figures reveals that the proposed SAMDE algorithm yields a better approximation of the desired response as compared to works presented in [5], [11], [12], [16] and [17]. The ripple in the stop-band of Fig.3(a) is much less as compared to Fig.3(b)∼(f).



(a) SAMDE method

(b) Lp technique in [5]

(c) Neural Network method in [12]

(d) GA-based method in [13]

(e) MEPSO method in [17]

(f) TBIA method in [18]

Fig. 3. Comparison of the amplitude response $|M(\omega_1, \omega_2)|$ of the considered 2-D filter.

## 6. Conclusions

In this paper, the SAMDE approach has been presented to design the 2-D recursive digital filter. The SAMDE approach adopts an adaptive mutation operator that combines with the advantages of the DE/rand/1/bin strategy and the DE/best/2/bin strategy. This new mutation strategy can keep the balance between the global exploration and local exploitation, and then improve the convergence rate and accuracy of the DE algorithm. In the studied problem of designing a 2-D recursive digital filter, there are many parameters and numerous local optimal and that are challenging enough for evaluating the performance of the proposed SAMDE approach and performance of previous approaches. A specific example of the design problem shows that compared to the methods of NN in [11], GA in [12], MEPSO in [16] and TBIA in [17], which, to our knowledge, are the most recent and the best-known methods to date, the proposed SAMDE approach can yield a better approximation of the desired response. Furthermore, the filter obtained has a reasonably good stability margin since we introduce the desired stability as appropriate constraints.

## Acknowledgements

The authors would like to thank the reviewers for their useful comments for improving the paper quality. This work was partly supported by the National Natural Science Foundation of China (No.60835004, 60974048), National High Technology Research and Development Program of China (2008AA04Z214), Natural Science Foundation of Hunan Province (No.09JJ9012), and Aid Program for Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province.

## References

1. T. Kaczorek, *Two-Dimensional Linear Systems*, Springer, Berlin, 1985.
2. A. Antoniou, *Digital Signal Processing: Signals, Systems, and Filters*, McGraw-Hill, NewYork, 2006.
3. C. L. Chan and H. K. Kwan, “Simple design of 2-dimensional circularly symmetric recursive digital filters using 1-dimensional analogue filters,” *International Journal of Electronics*, **67**, 585–590 (1989).
4. T. Williams, M. Ahmadi and W. C. Miller, “Design of 2D FIR and IIR digital filters with canonical signed digit coefficients using singular value decomposition and genetic algorithms,” *Circuits, Systems & Signal Processing*, **26**, 69–89 (2007).
5. W. S. Lu and A. Antoniou *Two-Dimensional Digital Filters*, Marcel Dekker, New York, 1992.
6. W. S. Lu and T. Hinamoto, “Optimal design of IIR digital filters with robust stability using Conic-Quadratic-Programming updates,” *IEEE Transactions on Signal Processing*, **51**, 1581–1592 (2003).
7. C. H. Hsieh, C. M. Kuo and Y. D. Jou, et al, “Design of two-dimensional FIR digital filters by a two-dimensional WLS technique,” *IIEEE Transactions on Circuits and Systems*, **44**, 348–412 (1997).
8. M. Daniel, A. Willsky, “Efficient implementations of 2-D noncausal IIR filters,” *IEEE Transactions on Circuits and Systems*, **44**, 549–563 (1997).
9. W. P. Zhu, M. O. Ahmad, M. N. S. Swamy, “A closed-form solution to the least-square design problem of 2-D linear-phase FIR filters,” *IEEE Transactions on Circuits and Systems*, **44**, 1032–1039 (1997).
10. B. Dumitrescu, “Optimization of two-dimensional IIR filters with nonseparable and separable denominator,” *IEEE Transactions on Signal Processing*, **53**, 1768–1778 (2005).
11. V. Mladenov, N. Mastorakis, “Design of two-dimensional recursive filters by using neural networks,” *IEEE Transactions on Neural Networks*, **12**, 585–590 (2001).
12. N. Mladenov, F. I. Gonos, M. N. S. Swamy, “Design of two-dimensional recursive filters using genetic algorithms,” *IEEE Transactions on Neural Networks*, **50**, 634–639 (2003).
13. J. T. Tsai, J. H. Chou, T. K. Liu, et al, “Design of two-dimensional recursive filters by using a novel genetic algorithm,” in *IEEE International Symposium on Circuits and System*, Nashville, 2603–2606 (2005).
14. S. T. Tzeng, “Design of 2-D FIR digital filters with specified magnitude and group delay responses by GA approach,” *Signal Processing*, **87**, 2036–2044 (2007).
15. W. Fang, J. Sun, W. B. Xu, “Design of two-dimensional recursive filters by using quantum-behaved particle swarm optimization,” *Proc. of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Nashville, 2603–2606(2006).
16. S. Das, A. Konar, “A swarm intelligence approach to the synthesis of two-dimensional IIR filters,” *Engineering Applications of Artificial Intelligence*, **20**, 1806–1096 (2007).

17. J. T. Tsai, W. H. Ho, J. H. Chou, "Design of two-dimensional recursive filters by using Taguchi-based immune algorithm," *IET Signal Processing*, **2**, 110–117 (2008).
18. R. Storn, K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over Continuous Spaces," *Journal of Global Optimization*, **11**, 341–359 (1997).
19. K. Price, "Differential evolution vs. the functions of the 2nd ICEO," *IEEE International Conference on Evolutionary Computation*, Indianapolis, 153–157 (1997).
20. J. Vesterstrom, R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," *Proc. of the IEEE Congress on Evolutionary Computation (CEC'2004)*, Portland, Oregon, 1980–1987 (2004).
21. A. K. Qin, V. L. Huang and P. N. Sugannthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolution Computation*, **13**, 398–417 (2009).
22. M. Ujjwal, S. Indrajit, "Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery," *Pattern Recognition*, **42**, 2135–2149 (2009).
23. A. Qing, "Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems," *IEEE Transactions on Geoscience and Remote Sensing*, **44**, 116–125 (2006).
24. L. H. Wu, Y. N. Wang and S. W. Zhou, "Self-adapting control parameters modified differential evolution for trajectory planning manipulator," *Journal of Control Theory and Applications*, **5**, 365–374 (2007).
25. S. C. Leandro, C. M. Viviana, "Combing of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," *IEEE Transactions on Power Systems*, **21**, 989–996 (2006).
26. Z. Fan, J. Liu and S. Torben, "Improved differential evolution based on stochastic ranking for robust layout synthesis of MEMS components, *IEEE Transactions on Industrial Electronics*, **56**, 937-948 (2009).
27. Z. K. Deb, "An efficient constraint handling method for genetic algorithms" *Computer Methods in Applied Mechanics and Engineering*, **186**, 311–338 (2000).