ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **15**, Number 1, 2012, 50–62

A Modified Estimation of Distribution Algorithm for Digital Filter Design

Yuquan LI[1], Gexiang ZHANG[*1], Jixiang CHENG[1],
Xiangxiang ZENG[2], Marian GHEORGHE[3], Susan ELIAS[4]

[1] School of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, P. R. China
E-mail: wolf-quan@qq.com, {zhgxdylan, chengjixiang0106}@126.com
[2] Department of Computer Science, Xiamen University, Xiamen 361005, China
[3] Department of Computer Science, The University of Sheffield, Sheffield S1 4DP, UK
[4] Department of Computer Science & Engineering, SVCE, Chennai 602105, India
E-mail: [2]m.gheorghe@dcs.shef.ac.uk, [3]xzeng@xmu.edu.cn, [4]susanelias70@gmail.com

Abstract. Estimation of Distribution Algorithms (EDAs) are a class of probabilistic model-building evolutionary algorithms, which are characterized by learning and sampling the probability distribution of the selected individuals. This paper proposes a modified EDA (mEDA) for digital filter design. mEDA uses a novel sampling method, called centro-individual sampling, and a fuzzy C-means clustering technique to improve its performance. Extensive experiments conducted on a set of benchmark functions show that mEDA outperforms seven algorithms reported in the literature, in terms of the quality of solutions. Four types of digital infinite impulse response (IIR) filters are designed by using mEDA and the results show that mEDA can obtain better filter performance than four state-of-the-art methods.

Key-words: Modified EDA, Centro-individual sampling, Fuzzy C-means clustering, Digital IIR filter, Numeric optimization.

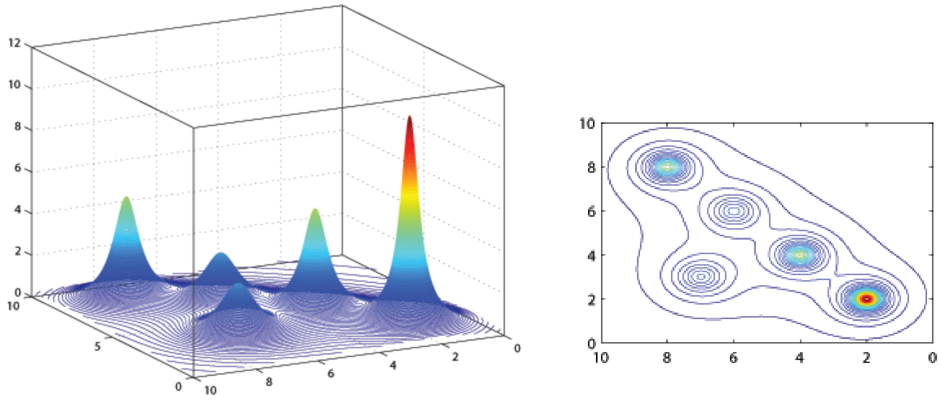
1. Introduction

Estimation of Distribution Algorithms (EDAs), introduced in 1996 [1], are a class of probabilistic model-building evolutionary algorithms. Differing from genetic algorithms (GAs), EDAs employ a probability distribution, built by using some selected individuals, to represent a population or a set of candidate solutions. EDAs use a statistical learning and probabilistic sampling method to generate offspring [2]. Nowadays EDAs attract much attention and are widely used in the community of evolutionary computation [3, 4, 12–15, 18, 29] and practical applications [17, 22]. Until now many variants of EDAs, such as PBILc [18], UMDAc [12] and IDEA [3] have been presented in the literature. As usual they assume that the distribution of variables satisfies a specific probabilistic model, such as Gaussian [14], Cauchy [6] or mixed distributions [22].

In EDAs, the most important point is to construct an appropriately probabilistic model based on the current population to generate offspring of the next population. In the early version of EDAs, a single Gaussian model was applied to establish the probabilistic distribution of some solutions selected [12]. They can obtain good solutions in solving unimodal and simple multimodal numeric optimization problems, whereas they have difficulties in solving complex multimodal optimization problems. The primary reason is that a single Gaussian model cannot simultaneously specify multiple probability distributions in complex multimodal problems. Thus this type of algorithm suffers seriously from a premature convergence which trapped in a suboptimal solution. To overcome the drawbacks, many researchers introduced other techniques into EDAs for improving their performance. In [14], "Clustering and Estimation of Gaussian Network Algorithm based on BGe metric" ($\text{CEGNA}_{\text{BGe}}$) and "Clustering and Estimation of Gaussian Distribution Algorithm" (CEGDA) were presented to solve multimodal functions with several local optima by incorporating a clustering technique with EDAs. In [11], a niche separation was introduced into two variants of EDAs, MIMIC and EBNA, and experimental results show the introduction of the niche separation can improve the MIMIC and EBNA performance. In [7], NichingEDA was proposed by combining a niche method and a recombination operator with an EDA.

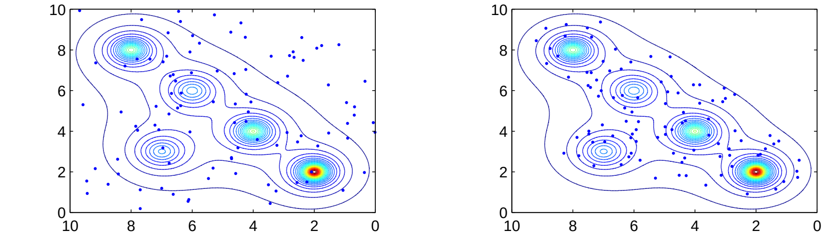
In EDAs reported in the literature, a group or all of the individuals in a population were used to construct a probability distribution for samples to produce offspring. Thus they mainly focused on the global information of a group or a population. In fact, at the former stage of searching solutions of complex multimodal optimization problems, the information of each of individuals is also important for building an appropriately probabilistic model to produce the individuals scattered across the whole search space at the next generation; while at the latter stage, the clustering of individuals is useful to make different groups of individuals focus on different regions related to various peaks of the problem.

This paper proposes a modified EDA (mEDA) to design digital IIR filters by introducing a novel sampling method, called centro-individual sampling (CIS) and a fuzzy C-means clustering technique (FCMC). Extensive experiments conducted on 27 benchmark test problems and 4 typical types of IIR filters show that mEDA is competitive to HPBILc, UMDA_c^G, $\text{EMNA}_{\text{global}}$, EEDA, $\text{CEGNA}_{\text{BGe}}$, CEGDA, NichingEDA, HGA, HTGA, TIA, and CCGA.



(a) Landscape in 3-D dimension (b) Landscape in 2-D dimension

Fig. 1. Landscapes of Shekel function.



(a) Distribution of an initial population (b) Population distribution after some generations

Fig. 2. Changes of population distributions in the process of evolution.

2. mEDA

According to investigations [4,10,14,26], a premature convergence of EDAs usually results from two aspects: 1) the variance of probabilistic models decreases at an exponential rate; 2) the process of searching better solutions is usually performed in the neighborhood of the center of the selected individuals, which is not appropriate for maintaining population diversity. To overcome the shortcomings, this paper uses CIS and FCMC to improve the EDA performance.

Evolutionary algorithms (EAs) often involve two steps: population initialization and population evolution. At the early stage of evolution, EAs search for candidate solutions in the whole decision space. As the evolution proceeds, the individuals in the worse fitness area will be washed out, while the survived individuals will cluster in the better fitness area. The Shekel function with two decision variables is used as an example to illustrate the idea. Figure 1 shows the landscape of the Shekel function, where the problem has five peaks in the decision space and the global optimum is located at (2,2). The distribution of an initial population is shown in Fig. 2(a), where the individuals evenly distribute in the decision space. However, after a certain number of evolutionary generations, the distribution changes and is shown in Fig. 2(b), where individuals form several groups scattering around five peaks. This observation motivates us to divide the population into several clusters, each of which can move forward an extrema. Thus all the individuals cannot jump down to one extremum and finally a better solution can be obtained. This is the main motivation of the proposal of mEDA.

In mEDA, a splitting ratio R_s, $R_s = t/G_m$, is used for determining the moment when the population is divided into several clusters, where G_m is the maximal number of generations. When $t < R_s \cdot G_m$, all of individuals follow one specific distribution. Thus the standard deviation of the fitness values of all individuals is large enough to produce offspring covering the whole space. When $t \geq R_s \cdot G_m$, we use FCMC to classify the population into N_c clusters, each of which evolves independently. The pseudocode algorithm of mEDA is shown in Fig. 3.

```
input:   Popsize, G_m, R_s, N_c
initialization
for  t = 1 : G_m
   if  t < R_s•G_m
         perform CIS on pop(t)
         update population by conducting pair-wise selection
   elseif  t ≥ R_s•G_m
         perform FCMC to get N_c clusters
         perform CIS on each cluster independently
         update population in each cluster with pair-wise selection
   end
end
output: the best result
```

Fig. 3. Pseudocode algorithm of mEDA.

In what follows we discuss CIS. As usual EDAs use the mean of selected individuals as the center of a probability distribution to produce offspring. So the probability distribution built is usually in the neighborhood of the center of all individuals. This approach easily results in the loss of population diversity and therefore the loss of further exploration capability and trapping in a local optima. To overcome the drawbacks, mEDA introduces CIS to establish a probability distribution for each of parent individuals to generate one child. To be specific, each of the parent individuals is used as the center of a probability distribution to produce offspring. Thus we can generate an individual in the neighborhood of each parent individual. CIS is helpful to make full use of the information of each individual to maintain population diversity.

In the univariate Gaussian distribution, the probability density function of the variable in the *j-th* dimension is $p_j(x) = N(u_j, \sigma_j)$, where u_j and σ_j are the mean value and standard deviation of the variable in the *j-th* dimension of all individuals in a population, respectively. Thus the sampling probabilistic model in mEDA is built. To obtain better population diversity, mEDA uses CIS to generate offspring. Assuming that $X_j = (x_{ji}|i = 1, 2, \cdots, S)$, $j = 1, 2, \cdots, N$, is a selected individual, where N and S are the population size and the number of dimensions, respectively, and the standard deviation is $\sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_s\}$, we generate offspring by using the formula

$$x_{ji}^{new} = x_{ji} + S_e \cdot N(0, \sigma_i) \tag{1}$$

where S_e is a scale and is set to 0.5 in this paper; x_{ji}^{new} is a new individual generated. The pseudocode algorithm of CIS is shown in Fig. 4.

```
intput: a parent population pop, the scale Se and
    an offspring population size popsize
step 1: estimate the distribution of pop, obtained
    the standard deviation of pop
step 2: generate offstring using (1)
    with popsize individuals
step 3: boundary checking
output: offstring population
```

Fig. 4. Pseudocode algorithm of CIS.

To clearly illustrate the differences between CIS and the approach used widely in EDAs, we use a 2-dimensional problem as an example, which is shown in Fig. 5, where (a)-(d) show a Gaussian density function, the distribution of individuals in the parent population, the distribution of individuals in the offspring population obtained by using the usual sampling methods and the distribution of individuals in the offspring population obtained by using CIS, respectively. It is clear that CIS can generate individuals with a much wider distribution than the usual methods. Thus CIS can be more beneficial to population diversity than the usual methods.

In the following description, we discuss FCMC, which is applied to partition a data set into c fuzzy groups through minimizing a dissimilarity measure [1, 2]. Let $\{x_i|i = 1, \cdots, n\}$ be a data set, c be the number of clusters and $\{m_j|j = 1, 2, \cdots, c\}$ be the center of each cluster, u_{ij} (s.t. $\sum_{j=1}^{c} u_{ij} = 1, u_{ij} \in [0, 1], \forall i = 1, \cdots, n$) be the membership element which represents the degree of membership between x_i and the *j-th* cluster. The objective function for FCMC can be denoted as

$$J = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^{b} d_{ij}^{2} \tag{2}$$

where b is a weighting exponent and usually set to 2; d_{ij} is the Euclidean distance between *j-th* cluster center and *i-th* data points. $u_{ij} = 1/\sum_{k=1}^{c} \left(\frac{d_{ik}}{d_{ij}}\right)^{2/(b-1)}$. The cluster center of the *j-th* fuzzy group can be denoted as $m_j = \sum_{i=1}^{n} u_{ij}^b x_i / \sum_i^n u_{ij}^b$. The cluster center and degree of membership need to be updated. The detailed steps of FCMC can refer to [2].

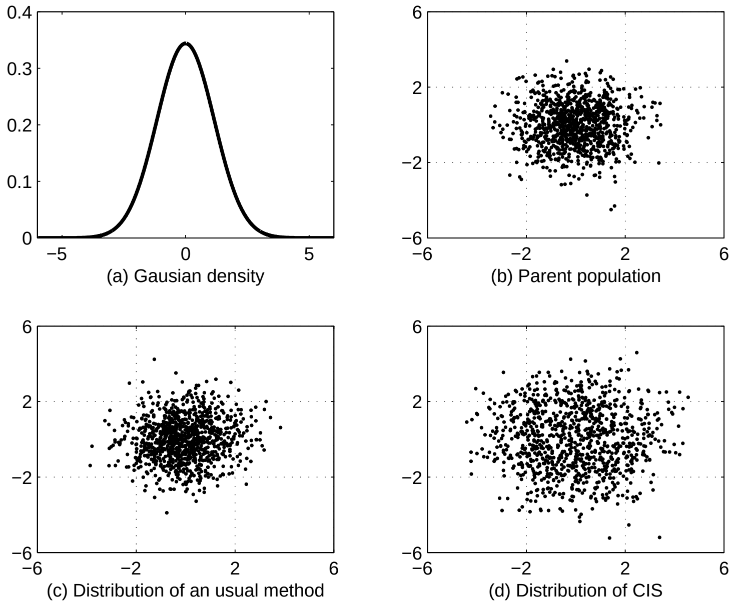


Fig. 5. Differences between CIS and the usual sampling methods.

3. Numerical Experiments

A test suit, which is widely used in the literature, is employed to test the mEDA performance. The test suit consists of unimodal and multimodal problems abstracted from [24] and [7].

3.1. Comparison with HPBILc

The 23 functions from [24], denoted as $f_1, f_2, \cdots, f_{23}$, are applied to conduct the experiments. Functions $f_1 - f_{13}$ are high-dimensional problems, where $f_1 - \mathrm{f}_7$ are unimodal functions and $f_8 - \mathrm{f}_{13}$ are multimodal functions with many local optima. Functions $f_{14} - \mathrm{f}_{23}$ are low-dimensional problems with a few local optimum. The details of these problems are referred to [23]. According to empirical studies, the parameters of mEDA are set as follows: the population size for $f_1 - f_3, f_9$ and the rest problems is set to 50, 100 and 200, respectively; the number of clusters is set to 3 and 5 for smaller and more than 100 individuals in the population, respectively; R_s is assigned as 0.25; the number of function evaluations in [23] for each problem is used as stopping condition. Thirty independent runs are performed for each of problems. Experimental results are listed in Table 1, where the mean values (mv) and standard deviations (sd), represented as mv±sd, are computed on the best solutions found over 30 independent runs. In Table 1, the statistical results of HPBILc in [23] are also provided. HPBILc is an EDA for continuous optimization based on histogram probabilistic model, which utilizes the sub-dividing strategy to guarantee the accuracy of optimal solutions.

Table 1. Experimental results of mEDA and HPBILc on 23 test functions (Mean values (mv) and standard deviations (sd) are organized as mv±sd.)

Functions		mEDA (mv±sd)	HPBILc (mv±sd)
Unimodal functions	f_1	**1.84e-16±1.01e-15**	3.82e-08±9.83e-09
	f_2	**1.82e-27±1.64e-27**	1.93e-09±1.51e-10
	f_3	7.89e-06±**1.50e-20**	**8.59e-19**±5.16e-19
	f_4	**5.72e-05±2.51e-05**	1.70e-01±1.90e-01
	f_5	**1.50e-01**±5.96e-01	6.42e-00±**3.10e-01**
	f_6	**0.00e-00±0.00e-00**	**0.00e-00±0.00e-00**
	f_7	**2.47e-27±3.04e-27**	3.56e-03±8.39e-04
High-dimension multimodal functions	f_8	-1.20e+04±2.29e+02	**-1.26e+04±9.42e-08**
	f_9	3.40e+01±1.26e+01	**0.00e-00±0.00e-00**
	f_{10}	**3.25e-09±1.07e-09**	2.75e-05±3.37e-06
	f_{11}	**1.32e-14**±1.05e-14	5.95e-14±**1.03e-14**
	f_{12}	**3.32e-17±2.87e-17**	1.54e-08±6.12e-09
	f_{13}	**3.29e-14±1.80e-13**	1.50e-06±8.38e-07
Low-dimension multimodal functions	f_{14}	**9.98e-01±6.68e-008**	1.00e-00±4.10e-03
	f_{15}	**3.34e-04**±6.03e-05	7.74e-04±**1.39e-05**
	f_{16}	**-1.03e-00±7.10e-09**	**-1.03e-00**±5.54e-05
	f_{17}	3.99e-01±**1.45e-10**	**3.98e-01**±1.09e-06
	f_{18}	**3.00e-00±6.48e-06**	**3.00e-00**±5.20e-03
	f_{19}	**-3.86e-00±1.66e-011**	**-3.86e-00**±4.95e-06
	f_{20}	**-3.32e-00**±1.88e-02	**-3.32e-00±1.78e-03**
	f_{21}	**-9.87e-00±2.80e-01**	-9.65e-00±3.6e-01
	f_{22}	**-10.01e-00**±5.00e-01	-9.81e-00±**3.5e-01**
	f_{23}	**-10.36e-00±1.80e-01**	-9.84e-00±2.70e-01

It can be seen from Table 1 that mEDA performs better than HPBILc, in terms of mean solutions and the robustness of solutions. According to the studies in [9], Wicoxon's and Friedman's tests are appropriate for analyzing the behavior of optimization algorithms with respect to multiple optimization problems. We employ the two statistical tests to check whether there is a significant difference between mEDA and HPBILc. In the experiment, 0.05 is considered as the level of significance. Experimental results show that the values of Wicoxon's test and Friedman's test are 0.0395 and 0.0016, respectively, which clearly indicates that mEDA really outperforms HPBILc.

3.2. Comparisons with EDAs based on Gaussian model

In this subsection, 5 test problems, f_1, f_5, f_{10}, f_{14} and f_{15} abstracted from [24], are used to compare the mEDA performance with $\mathrm{UMDA}_\mathrm{c}^\mathrm{G}$ [12], $\mathrm{EMNA}_\mathrm{global}$ [12], EEDA [12] and NichingEDA [7]. Four complex multi-modal problems, f_{24} (Two-peaks function), f_{25} (ThreePeaks function), f_{26} (Shekel function) and f_{27} (Schwefel) used in [7] are employed to draw a comparison between mEDA and $\mathrm{CEGNA}_\mathrm{BGe}$ [14], CEGDA [14] and NichingEDA. $\mathrm{UMDA}_\mathrm{c}^\mathrm{G}$, $\mathrm{EMNA}_\mathrm{global}$ and EEDA are three EDAs with a single classical Gaussian model, but they employ different model hypothesis. $\mathrm{CEGNA}_\mathrm{BGe}$ and CEGDA use a clustering technique to deal with multi-modal functions. NichingEDA adopts a niching method and recombination operators to solve the traditional "EDA-hard problems". To bring a fair comparison with $\mathrm{UMDA}_\mathrm{c}^\mathrm{G}$, $\mathrm{EMNA}_\mathrm{global}$ and EEDA without any clustering technique, in the experiments of mEDA for the five problems, f_1, f_5, f_{10}, f_{14} and f_{15}, the parameters N_c, R_s and the population size are set to 1, 1 and 100, respectively. The algorithm stops when the maximal number of generations reaches 5000, which is the same in [7]. The statistical results of mEDA over 30 independent runs are listed in Table 2. The results of $\mathrm{UMDA}_\mathrm{c}^\mathrm{G}$, $\mathrm{EMNA}_\mathrm{global}$, EEDA and NichingEDA come from [7]. In the experiments of mEDA for functions f_{24}, f_{25}, f_{26} and f_{27}, the population size, N_c, R_s and the maximal number generations are assigned as 100, 5, 0.25 and 4000, respectively. The statistical results of mEDA over 30 independent runs are given in Table 3, where the results of three algorithms in [7], $\mathrm{CEGNA}_\mathrm{BGe}$, CEGDA and NichingEDA, are also listed.

Table 2. Experimental results of $\mathrm{UMDA}_\mathrm{c}^\mathrm{G}$, $\mathrm{EMNA}_\mathrm{global}$, EEDA, NichingEDA and mEDA on 5 test functions, f_1, f_5, f_{10}, f_{14} and f_{15} (Standard deviations are put in the parentheses and D represents the number of dimensions of each problem.)

Functions	$\mathrm{UMDA}_\mathrm{c}^\mathrm{G}$	$\mathrm{EMNA}_\mathrm{global}$	EEDA	NichingEDA	mEDA
f_1	1.6529e-043	1.0708e-039	8.6746e-044	4.5074e-009	**6.7635e-113**
(D=10)	(5.0208e-044)	(3.1585e-040)	(2.6313e-044)	(9.3440e-009)	**(8.0028e-113)**
f_5	8.1958e-000	7.7806e-000	6.3004e-000	4.6258e-000	**6.2840e-001**
(D=10)	**(3.7400e-002)**	(1.7130e-001)	(1.2590e-001)	(1.6986e-000)	(1.0217e-000)
f_{10}	**4.4409e-016**	**4.4409e-016**	**4.4409e-016**	2.4350e-005	**4.4409e-016**
(D=10)	**(0.0000e-000)**	**(0.0000e-000)**	**(0.0000e-000)**	(1.3137e-005)	**(0.0000e-000)**
f_{14}	1.1691e-000	1.0747e-000	1.0019e-000	**9.9800e-001**	**9.9800e-001**
(D=2)	(3.4060e-001)	(2.543e-001)	(1.5600e-002)	**(0.0000e-000)**	**(0.0000e-000)**
f_{15}	2.6000e-003	2.0000e-003	1.1000e-003	3.0874e-004	**3.0749e-004**
(D=4)	(1.0000e-003)	(7.5969e-004)	(1.0183e-004)	(2.7899e-006)	**(1.3468e-019)**

From the results in Table 2, we can see that three EDAs, $\mathrm{UMDA}_\mathrm{c}^\mathrm{G}$, $\mathrm{EMNA}_\mathrm{global}$ and EEDA, which used a single Gauss distribution, obtain good solutions for f_1 and f_{10}, but they have difficulties in solving multimodal functions such as f_5, f_{14} and f_{15}. NichingEDA gets better solutions of f_5, f_{14} and f_{15} and worse solutions of f_1 and f_{10} than $\mathrm{UMDA}_\mathrm{c}^\mathrm{G}$, $\mathrm{EMNA}_\mathrm{global}$ and EEDA. The proposed mEDA is the best among the 5 approaches with respect to mean values and standard deviations of these functions except for the standard deviation of f_5 worse than $\mathrm{UMDA}_\mathrm{c}^\mathrm{G}$. These results show that mEDA has a good performance on both unimodal and mulitmodal functions. Functions f_{24}-f_{27} are difficult to solve due to many local optima. Table 3 shows that mEDA is competitive to $\mathrm{CEGNA_{BGe}}$, CEGDA and NichingEDA. Actually the three EDAs were designed for dealing with multimodal functions by taking advantage of clustering techniques or a niching method. Thus these results in Table 3 also indicate that mEDA is feasible in solving these complex multimodal optimization problems, which comes from the reasons that CIS is helpful to maintain population diversity and FCMC is beneficial to make the mEDA algorithm keep off the local optimum.

Table 3. Experimental results of $\mathrm{CEGNA_{BGe}}$, CEGDA, NichingEDA and mEDA on 4 problems, f_{24}, f_{25}, f_{26} and f_{27} (f_{24}-f_{26} are maximization problems and f_{27} is a minimization problem. Standard deviations are put in the parentheses and D represents the number of dimensions of each problem.)

Functions	$\mathrm{CEGNA_{BGe}}$	CEGDA	NichingEDA	mEDA
f_{24}	**10.1053e-000**	10.0999e-000	**10.1053e-000**	**10.1053e-000**
(D=5)	(3.5500e-015)	(5.9200e-003)	**(0.0000e-000)**	**(0.0000e-000)**
f_{25}	10.1053e-000	10.1048e-000	**10.1053e-000**	**10.1053e-000**
(D=5)	(3.5500e-015)	(7.9900e-004)	**(0.0000e-000)**	(3.6100e-015)
f_{26}	**10.0134e-000**	**10.0134e-000**	2.2300e-001	**10.0134e-000**
(D=30)	**(8.8818e-015)**	**(8.8818e-015)**	(5.6400e-000)	(2.2767e-011)
f_{27}	-6.7604e+003	-5.9225e+003	-8.0054e+003	**-1.1096e+004**
(D=30)	(2.6243e+003)	(1.8935e+003)	**(2.7055e+002)**	(3.9231e+002)

Table 4. Prescribed design requirements of four types of IIR filters

Filter type	Passband	Stopband	Maximal value of magnitude response
LP	$[0,\ 0.2\pi]$	$[0.3\pi,\ \pi]$	1
HP	$[0.8\pi,\ \pi]$	$[0,\ 0.7\pi]$	1
BP	$[0.4\pi,\ 0.6\pi]$	$[0.4\pi,\ 0.6\pi] \cup [0.75\pi,\ \pi]$	1
BS	$[0.4\pi,\ 0.6\pi] \cup [0.75\pi,\ \pi]$	$[0.4\pi,\ 0.6\pi]$	1

4. mEDA for IIR Filter Design

Digital IIR filter design is a multiparameter and multicriterion optimization problem [19]. How to design a good digital IIR filter is always a hot topic in digital signal processing [19–21, 25]. In the optimal design of a digital filter, the transfer function is usually formulated as

$$H(z) = G \prod_{i=1}^{N} \frac{z + a_{i1}}{z + b_{i1}} \prod_{j=1}^{M} \frac{z^2 + c_{j1}z + c_{j2}}{z^2 + d_{j1}z + d_{j2}}, \tag{3}$$

where G is the gain; $a_{i1}, b_{i1}(i = 1, 2, \cdots, N)$ are the first-order coefficients; and $c_{j1}, c_{j2}, d_{j1}, d_{j2}(j = 1, 2, \cdots, M)$ are the second-order coefficients. They are represented as a parameter set $X = \{(a_{11}, b_{11}, \cdots, a_{N1}, b_{N1}), (c_{11}, c_{12}, d_{11}, d_{12}, \cdots, c_{M1}, c_{M2}, d_{M1}, d_{M2}), G\}$. The task of an IIR filter design is to find a proper parameter set X which can produce the desired response and X must satisfy the stability constraints, $b_{i1} \in [-1, 1]$, $d_{j1} \in [-1 - d_{j2}, 1 + d_{j2}]$.

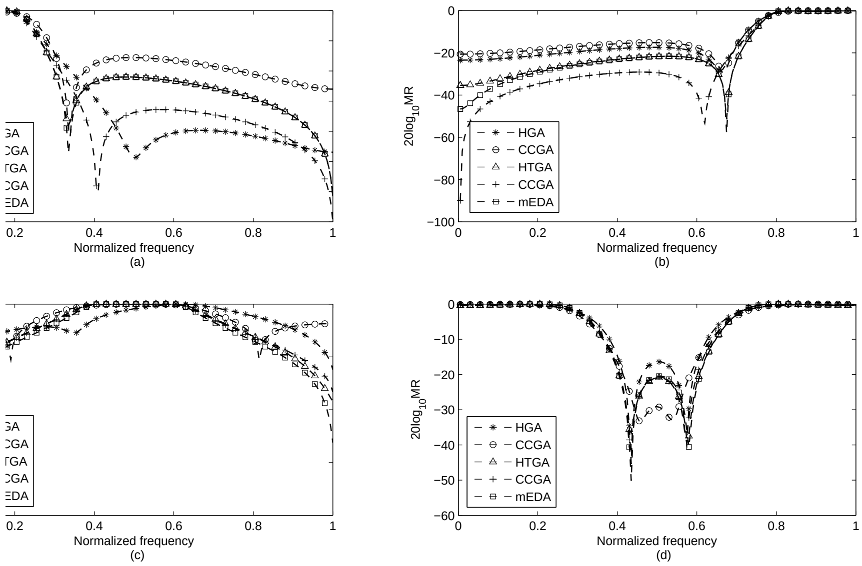


Fig. 6. Magnitude responses of IIR filters. (a) LP (b) HP (c) BP (d) BS.

To bring a comparison, four typical types of IIR filters (lowpass (LP), highpass (HP), bandpass (BP) and bandstop (BS)), the design criterion $\delta_1(x) + \delta_2(x)$ and the performance evaluation criteria (ripples in passband and stopband) in [19] are considered, where $\delta_1(x)$ and $\delta_2(x)$ are ripples in passband and stopband of filters, respectively. The prescribed design requirements of four types of IIR filters are given in Table 4. Several state-of-the-art methods, HGA [19], HTGA [20], TIA [21], CCGA [25], are used as benchmark algorithms for design digital IIR filters.

In the experiments of mEDA, the population size, the number of clusters, R_s and the maximal number of generations as stopping condition are assigned as 100, 3, 0.25, and 400, respectively. The four filters designed by mEDA are as follows:

$$H_{LP}(z) = 0.085929 \frac{(z + 0.999387)(z^2 - 1.023525z + 1.014613)}{(z - 0.571704)(z^2 - 1.347394z + 0.745515)} \tag{4}$$

$$H_{HP}(z) = 0.093000 \frac{(z - 0.919411)(z^2 + 1.019735z + 1.004044)}{(z + 0.563953)(z^2 + 1.339855z + 0.742839)} \tag{5}$$

$$\begin{aligned} H_{BP}(z) =& 0.036399 \frac{(z^2 + 0.023022z - 1.015411)(z^2 - 0.008149z - 0.994221)}{(z^2 + 0.782489z + 1.267838)(z^2 - 0.782612z + 1.267805)} \\ & \times \frac{(z^2 + 0.016250 - 0.978428)}{(z^2 - 0.000035 + 0.576865)} \end{aligned} \tag{6}$$

$$H_{BP}(z) = 0.413806 \frac{(z^2 - 0.453903z + 1.012873)(z^2 + 0.471868z + 1.021520)}{(z^2 - 0.852523z + 0.548847)(z^2 + 0.863515z + 0.552537)} \quad (7)$$

The Magnitude responses (MR) and the ripples in the passband and stopband of the filters designed by mEDA, HGA, HTGA, TIA, and CCGA are shown in Fig. 6 and Table 5, respectively. In Table 5, MRP and MRS denote the maximal ripples in the passband and stopband, respectively. Figure 6 and Table 5 indicate the following conclusions: (1) mEDA obtains much smaller values of the ripples in the passband and stopband than HGA and CCGA, and a little bit better results than HTGA and TIA; (2) mEDA has a good performance with respect to the four different types of digital IIR filters. Digital IIR filter design can be viewed as a multi-parameter and multi-criterion optimization problem with many local optima. mEDA has a good performance in solving multimodal optimization problems with lots of local extrema, which has been demonstrated in Section 3, so it can obtain good solutions in the design of digital IIR filters.

Table 5. The performance of IIR filters designed by five methods

Filters	methods	$\delta_1(x) + \delta_2(x)$	MRP	MRS
LP	HGA	0.2938	**0.0138**	0.1800
	HTGA	0.1214	0.0414	0.0800
	CCGA	0.2635	0.0966	0.1669
	TIA	0.1213	0.0437	**0.0776**
	mEDA	**0.1212**	0.0414	0.0798
HP	HGA	0.2598	0.0799	0.1819
	HTGA	0.1221	0.0388	0.0833
	CCGA	0.2705	0.0956	0.1749
	TIA	0.1990	0.0533	0.1457
	mEDA	**0.1210**	**0.0381**	**0.0829**
BP	HGA	0.2816	0.1044	0.1772
	HTGA	0.0945	0.0234	0.1711
	CCGA	0.2734	0.1080	0.1654
	TIA	0.0910	0.0226	0.0685
	mEDA	**0.0654**	**0.0214**	**0.0440**
BS	HGA	0.2806	0.1080	0.1726
	HTGA	0.1411	0.0455	0.0956
	CCGA	0.2768	0.1035	0.1733
	TIA	0.1411	0.0540	0.1171
	mEDA	**0.1393**	**0.0439**	**0.0954**

5. Conclusion

This paper presents a modified EDA, which uses a novel sampling method, CIS, and a fuzzy C-means clustering technique, FCMC, to improve the performance of EDAs. CIS is helpful to maintain population diversity and FCMC is beneficial to make the algorithm keep off local optima. Extensive experiments are conducted by using several global optimization problems with high dimensions and several optimization problems with simple landscapes or irregular landscapes. Experimental results show that mEDA is competitive to several types of EDAs, such as HPBILc, CEGDA, CEGNABGe and NichingEDA, in terms of the quality of solutions. Further, in the design of digital IIR filters, which are considered as multi-parameter and multimodal optimization problems, mEDA also obtain better filter performance than four approaches, HGA, HTGA, TIA and CCGA, reported in the literature, in terms of four typical frequency-selective filters, lowpass, highpass, bandpass and bandstop filters. Our further work will focus on using a proper EDA to design digital IIR filters, which will be formulated as multi-objective optimization problems by considering contrary performance criteria for different bands of filters.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (61170016), the Program for New Century Excellent Talents in University, the project-sponsored by SRF for ROCS, SEM, the Scientific and Technological Funds for Young Scientists of Sichuan (09ZQ026-040), the Fund for Candidates of Provincial Academic and Technical Leaders of Sichuan and the Fundamental Research Funds for the Central Universities (SWJTU11ZT07).

References

[1] Baraldi A., Blonda P., *A survey of fuzzy clustering algorithms for pattern recognition, I,* IEEE Trans. SMCB, **29**, pp. 778–785, 1999.

[2] Bezdek J. C., Ehrlich R., Full W., *FCM: The fuzzy C-means clustering algorithm,* Computers and Geosciences, **10**, pp. 191–203, 1984.

[3] Bosman P., Thierens D., *Expanding from discrete to continuous estimation of distribution algorithms: The IDEA*, Parallel Problem Solving from Nature, **1917**, pp. 767–776, 2000.

[4] Bosman P. A. N., Grahl P., Rothlauf P., *SDR: a better trigger for adaptive variance scaling in normal EDAs,* Proc. GECCO, pp. 492–499, 2007.

[5] Bosman P., Grahl J., Thierens D., *Enhancing the performance of maximum-Likelihood gaussian EDAs using anticipated mean shift,* Parallel Problem Solving from Nature, **5199**, Springer, pp. 133–143, 2008.

[6] Carrillo R. E., Aysaft T. C., Barner K. E., *Generalized cauchy distribution based robust estimation*, Proc. ICASSP, pp. 3389–3392, 2008.

[7] Dong W. S., Yao X., *NichingEDA: Utilizing the diversity inside a population of EDAs for continuous optimization*, Proc. CEC, pp. 1260–1267, 2008.

[8] Dong W. S., Yao X., *Unified eigen analysis on multivariate gaussian based estimation of distribution algorithms*, Information Sciences, **178**, pp. 3000–3023, 2008.

[9] García S., Molina D., Lozano M., Herrera F., *A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the CEC'2005 special session on real parameter optimization*, J. Heuristics, **15**, pp. 617–644, 2009.

[10] Grahl J., Bosman P. A. N., Rothlauf F., *The correlation-triggered adaptive variance scaling IDEA,* Proc. GECCO, pp. 397–404, 2006.

[11] Handa H., *Estimation of distribution algorithms with niche separation mechanism*, Proc. CEC, 2007, pp. 119–126.

[12] Larranaga P., Lozano J. A., *Estimation of distribution algorithms: A new tool for evolutionary computation*, Boston: Kluwer Academic Publishers, 2001.

[13] Larranaga P., Etxeberria R., Lozano J. A., Pena J. M., *Optimization in continuous domains by learning and simulation of gaussian networks*, Proc. GECCO, pp. 201–204, 2001.

[14] Lu Q., Yao X., *Clustering and learning gaussian distribution for continuous optimization*, IEEE Trans. SMCC, **35**, pp. 195–204, 2005.

[15] Mülenbein H., Paass G., *From recombination of genes to the estimation of distributions I. binary parameters*, Parallel Problem Solving from Nature, **1141**, Springer, pp. 178–187, 1996.

[16] Saeys Y., Degroeve S., Aeyels D., Van Y., Peer d., Rouz P., *Fast feature selection using a simple estimation of distribution algorithm: a case study on splice site prediction*, Bioinformatics, **19**, pp. 179–188, 2003.

[17] Sagarna R., Arcuri A., Yao X., *Estimation of distribution algorithms for testing object oriented software*, Proc. CEC, pp. 438–444, 2007.

[18] Sebag M., Ducoulombier A., *Extending population-based incremental learning to continuous search spaces*, Parallel Problem Solving from Nature, **1498**, p. 418, 1998.

[19] Tang K. S., Man K. F., Kwong S., Zhou Z. F., *Design and optimization of IIR filter structure using hierarchical genetic algorithms*, IEEE Trans. IE, **45**, pp. 481–487, 1998.

[20] Tsai J. T., Chou J. H., Liu T. K., *Optimal design of digital IIR filters by using hybrid taguchi genetic algorithm*, IEEE Trans. IE, **53**, pp. 867–879, 2006.

[21] Tsai J. T., Chou J. H., *Design of optimal digital IIR filters by using an improved immune algorithm*, IEEE Trans. SP, **54**, pp. 4582–4596, 2006.

[22] Wang Y., Li B., *A Self-adaptive mixed distribution based uni-variate estimation of distribution algorithm for large scale global optimization*, Nature-Inspired Algorithms for Optimisation, **193**, pp. 171–198, 2009.

[23] Xiao J., Yan Y.P., Zhang J., *HPBILc: A histogram-based EDA for continuous optimization*, Applied Mathematics and Computation, **215**, pp. 973–982, 2009.

[24] Yao X., Liu Y., Lin G., *Evolutionary programming made faster*, IEEE Trans. EC, **3**, pp. 82–102, 1999.

[25] Yang Y., Yu X., *Cooperative coevolutionary genetic algorithm for digital IIR filter design*, IEEE Trans. IE, **54**, pp. 1311–1318, 2007.

[26] Zhang Q. F., *On the convergence of a factorized distribution algorithm with truncation selection*, Complexity, **9**, pp. 17–23, 2004.

[27] Zhang Q. F., Zhou A. M., Jin Y. C., *RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm*, IEEE Trans. EC, **12**, pp. 41–63, 2008.

[28] Zhang G. X., Jin W. D., Jin F., *Multi-criterion satisfactory optimization method for designing IIR digital filters*, Proc. ICCT, pp. 1484–1490, 2003.

[29] Zhang G. X., *Quantum-inspired evolutionary algorithms: a survey and empirical study*, J. Heuristics, **17**, pp. 303–351, 2011.