Hindawi Publishing Corporation The Scientific World Journal Volume 2014, Article ID 597278, 14 pages http://dx.doi.org/10.1155/2014/597278



# Research Article

# A Fast Elitism Gaussian Estimation of Distribution Algorithm and Application for PID Optimization

# Qingyang Xu, Chengjin Zhang, and Li Zhang

School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai 264209, China

Correspondence should be addressed to Qingyang Xu; xuqy1981@163.com

Received 27 January 2014; Accepted 9 March 2014; Published 27 April 2014

Academic Editors: P. Agarwal and Y. Zhang

Copyright © 2014 Qingyang Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Estimation of distribution algorithm (EDA) is an intelligent optimization algorithm based on the probability statistics theory. A fast elitism Gaussian estimation of distribution algorithm (FEGEDA) is proposed in this paper. The Gaussian probability model is used to model the solution distribution. The parameters of Gaussian come from the statistical information of the best individuals by fast learning rule. A fast learning rule is used to enhance the efficiency of the algorithm, and an elitism strategy is used to maintain the convergent performance. The performances of the algorithm are examined based upon several benchmarks. In the simulations, a one-dimensional benchmark is used to visualize the optimization process and probability model learning process during the evolution, and several two-dimensional and higher dimensional benchmarks are used to testify the performance of FEGEDA. The experimental results indicate the capability of FEGEDA, especially in the higher dimensional problems, and the FEGEDA exhibits a better performance than some other algorithms and EDAs. Finally, FEGEDA is used in PID controller optimization of PMSM and compared with the classical-PID and GA.

# 1. Introduction

Various optimization problems exist in engineering and academic research, which expect to find the best solution. If the problems are conventional or linear, the common mathematical methods will be effective. However, if the problems are too complicated to the common methods, some heuristic algorithms will be considered. Evolutionary algorithms (EAs) are very popular heuristic optimization techniques in the recent years. EAs are general terms of several optimization algorithms that are inspired by the Darwinian theory of natural evolution. It has the capability of solving the complicated optimization problems with nonlinear, high dimension and non-continuous characteristics. The algorithms search the optimal solution from many possible solutions, and the genetic operators, which simulate the principle of natural genetic evolution, are used to update the individuals. By several iterations, the optimal solution will be obtained, such as the genetic algorithms (GAs) [1], evolution strategies (ES), differential evolution (DE) [2, 3], and the artificial immune system (AIS) [4, 5] and also swarm evolutionary algorithm like particle swarm optimization (PSO) [2, 6, 7].

Although these algorithms have applied success to solve kinds of optimization problems [8], there are some inherent drawbacks. For example, if the building blocks spread all over the solutions, the standard EAs have very poor performance [9]. The recombination operators ether breaks the building blocks frequently or do not mix them effectively.

In recent years, estimation of distribution algorithms (EDAs) have attracted a lot of attention. It was proposed by Miuhlenbein and Paaß [10] and emerged as a generalization of EAs for overcoming some problems of EAs, like building blocks broken, poor performance in high dimensional problems, and the difficulty of modeling the distribution of solutions. Sometimes the gene blocks are built based on simple selection and crossover operators are not effective enough to get optimum solution as the blocks may be broken in EAs [9, 11]. Compared with building blocks in EAs, EDAs do not use the crossover or mutation operator to update individuals [12]. Instead, they extract the global statistical information from the superiority individual of previous iteration and build the distribution probability model of solution for sampling new individuals [13]. It is the main advantages of EDAs compared with EAs that the search

process is guided by the probabilistic model with explanatory and transparent characteristics [14, 15]. The algorithms are based on the probabilistic models following two steps: (1) Statistics the information of selected individuals and establish the probability model and (2) generate new population by sampling the probability model. Therefore, the new offspring of EDAs is based on the probability distribution instead of performing recombination of individuals as EAs.

The type of probabilistic models used by EDAs and the methods employed to learn them may vary according to the characteristics of the optimization problem. Therefore, different EDAs have been proposed for discrete and continuous problems. In traditional EDAs, the individuals are encoded with binary strings inheritance from EAs. In the populationbased incremental learning (PBIL) algorithm [16], the individuals are encoded as fixed length binary strings. The population of solutions is updated by the probability vector, which is initially set to probability 0.5 for each position of the binary strings. For univariate marginal distribution algorithm (UMDA) [10], the frequencies of values on each position are computed according to the selected individuals, which are then used to generate the new population. The compact genetic algorithm (cGA) [17] updates the population according to the probability vector like the PBIL. However, unlike the PBIL, it modifies the probability vector according to the contribution of individuals.

In case of real-valued problems, there are some approaches to extend EDAs to other domains, such as mapping other domains to the domain of fixed-length binary strings and then mapping the solution back to the problem's original domains, or extend or modify the class of probabilistic models to other domains. This first approach might lead to undesirable limitations and errors on real-coded problems. For the second method, the normal pdf is commonly used in continuous EDAs to represent univariate or multivariate distributions. Therefore, some EDAs based on the Gaussian distribution have been designed. In the stochastic hillclimbing with learning by vectors of normal distributions [18], the population of solutions is modeled by a vector of mean of Gaussian normal distribution  $\mu_i$  for each variable. The standard deviation  $\sigma$  is stored globally and it is the same for all variables. After generating a number of new solutions, the mean  $\mu_i$  are shifted towards the best solutions and the standard deviation  $\sigma$  is reduced to make future exploration more specifically. Various ways of modifying the  $\sigma$  parameter have been exploited in [19]. Regularized estimation of distribution algorithms (RegEDA) [20] makes use of regularized model estimation in EDAs for continuous optimization. The regularization techniques can lead to more robust model estimation in EDAs. Continuous Gaussian estimation of distribution algorithm (CGEDA) [14] which is a kind of multivariate EDAs is proposed for real-coded problems. Gaussian data distribution and dependent individuals are two assumptions that are considered in CGEDA. In the algorithm, the joint distribution of promising solutions is used in every dimension of the problem. In literature [21, 22], an estimation of distribution algorithm with Gaussian process based on a subspaces method was proposed, which can reduce the computation of complex problems. A real-coded EDA using

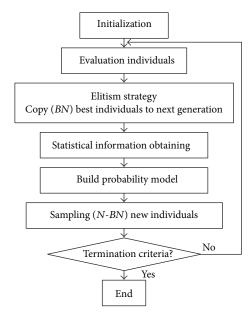


FIGURE 1: Flowchart of FEGEDA.

multiple probabilistic models (RMM) was proposed [23], which includes multiple types of probabilistic models with different learning rates and diversities. There are also other EDAs, which adopt more involved probability models and mixtures of pdfs. However, the probability models cannot reflect the problem completely, because it is hard to obtain an accurate model. In particular, with the increases of number of variables and the number of mixture components, the optimization results become unreliable [24]. Therefore, we specifically focus on the use of the single normal distribution in this paper, as it is more intuitive to be analyzed. Moreover, the use of single and easy normal pdf will not prevent us from obtaining a better understanding of the exploitation of the solutions. We propose a fast elitism Gaussian EDA (FEGEDA) based on the standard process of EDA. A fast learning rule is used to parameters of pdf learning, and an elitism strategy is used for a better performance. Hence, the increased convergence exhibited in this study is expected.

#### 2. The Fast Elitism Gaussian EDA

2.1. The Framework of the Algorithm. EDA is realized by probability estimation and sampling. The probability model is used to estimate the solution distribution, and the probability sampling is used to generate new individuals. In order to improve the performance of standard EDA, we adopt an elitism strategy in FEGEDA. Figure 1 is the flowchart of FEGEDA.

The steps of the FEGEDA are as follows.

Step 1 (initialization). Set the population size N, define the number BN of best individuals for probability model establishment and generate the initialized population Pop(0).

*Step 2* (population evaluation). Evaluate the *N* individuals  $x_1, x_2, \dots x_N$  according to fitness function f(x).

Step 3 (statistical information obtaining). Select *BN* best individuals according to the fitness and obtain the statistical information of mean  $\mu$  and standard deviation  $\sigma$ .

Step 4 (probability model  $P(x_1, x_2, ..., x_m)$  establishment). Use the fast learning rule and build the Gaussian normal distribution by the u of means and a covariance  $\sigma$ .

Step 5 (new population Pop(k) generation). Make use of sampling technique to generate a new population according to the probability model built in Step 4.

Step 6. Finally, the iteration is terminated according to the termination criteria. These criteria can be as simple as a fixed number of generations or imply a statistical analysis of the current population to evaluate the stopping condition criteria. If the stopping conditions do not meet, return to Step 2.

The probability model is built according to the distribution of the best solutions in the current population. Therefore, sampling solutions from this model should fall in promising areas with high probability. For some iterations, the solutions should be more likely to be close to the global optimum. The details of the main algorithm are explained in the following.

2.2. Initialization. In the algorithm, little parameters are needed to set except for the population size N and the best individuals size BN selected to build the probability model. Then, a random function is used to generate the initial population according to the variable domain  $[L_i, H_i]$ . Make use of random function generating variables  $z_i \in [a_i, b_i]$  and then convert to the domain  $[L_i, H_i]$  by

$$x_n^i = L_i + \frac{H_i - L_i}{b_i - a_i} (z_i - a_i),$$
 (1)

where  $x_n^i$  is the *i*th optimization variable of *n*th individual,  $z_i$  is the *i*th random variable,  $a_i$  and  $b_i$  are the bounds of *i*th random variable, and  $L_i$ , and  $H_i$  are the bounds of *i*th optimization variable.

2.3. Population Evaluation. In the individuals' evaluation, it depends on the characteristics of the problem. Conventionally, we should define an objective function f(x) in order to evaluate the fitness of individuals. Consider

Min (& Max) 
$$y = f(x)$$
  
S.t.  $g(x) = [g_1(x), g_2(x), ..., g_k(x)] \le 0$   
 $h(x) = [h_1(x), h_2(x), ..., h_j(x)] = 0$   
 $x = [x_1, x_2, ..., x_i, ..., x_m]$   
 $L_i \le x_i \le H_i \quad (i = 1, 2, ..., m),$  (2)

where x is m dimensional optimization variable, f(x) is the objective function,  $g_k(x)$  is the kth inequality constraints, and  $h_j(x)$  is the jth equality constraints.  $L_i$  and  $H_i$  are the bounds of variable.

2.4. The Establishment of Probability Model. The most important and crucial step of EDAs is the construction of probabilistic model for the solution distribution; to do this step of FEGEDA, Gaussian distribution of individuals is assumed to model and estimate the distribution of promising solutions in every dimension of the problem. Therefore, mean and standard deviation parameters of promising population are required which computed adaptively by maximum likelihood technique.

2.4.1. Statistical Information Acquisition. In order to construct a pdf model of the promising solutions, we should obtain the statistical information of promising solutions. Hence, statistical techniques have been extensively applied to the optimization problems. Fortunately, these parameters can be efficiently computed by the maximum-likelihood estimations [24].

In the pdf models that assume full independence, every variable is assumed independent of any variable. It must be noted that, in difficult optimization problems, different dependency relations can appear between variables and, hence, considering all of them independent may provide a model that does not represent the problem accurately. However, even if more involved probability models and mixtures of pdfs are defined and used in EDAs, the probability models cannot reflect the problem completely. For system modeling, the dependency relations between variables are very important. Conversely, for optimization problem, the problem decoupled as the combination of some independent variables is expected. Therefore, we specifically focus on the use of independent probability model to construct a fast elitism Gaussian EDA with better performance. That is, the probability distribution  $P(x_1, x_2, ..., x_m)$  of the vector  $(x_1, x_2, \dots, x_m)$  of m variables is assumed to consist of a product of the distributions of individual variables:

$$P(x_1, x_2, \dots x_m) = \prod_{i=1}^{m} P(x_i).$$
 (3)

This is very suitable for calculation. Different from the discrete case, the number of parameters to be estimated does not grow exponentially with *m*. Therefore, it is relatively fast and efficient.

The mean and covariance parameters of the normal pdf can be estimated from the selected individuals. Consider

$$\overline{\mu}_{i}(k) = \frac{1}{N} \sum_{n=1}^{BN} x_{i}^{n}(k),$$

$$\sigma_{i}^{2}(k) = \frac{1}{N} \sum_{n=1}^{BN} (x_{i}^{n}(k) - \overline{\mu}_{i}(k)) (x_{i}^{n}(k) - \overline{\mu}_{i}(k))^{T},$$
(4)

 $\overline{\mu}_i(k)$  is the mean of *i*th variable in *k*th iteration, *BN* is the selected individuals size, and  $\sigma_i^2(k)$  is the covariance of *i*th variable in *k*th iteration.

These parameters are always learned in the process of optimization. The iterative learning approaches are used in some literatures [23, 25–27] as follows:

$$\overline{\mu}_{i}(k) = \alpha \overline{\mu}_{i}(k) + \beta \overline{\mu}_{i}(k-1), \qquad (5)$$

$$\sigma_i^2(k) = \alpha \sigma_i^2(k) + \beta \sigma_i^2(k-1), \qquad (6)$$

where  $\alpha$  and  $\beta$  are the weights of  $\overline{\mu}_i(k)$  and  $\overline{\mu}_i(k-1)$ . The learning method depends on the class of models used; this step involves parametric or structural learning, also known as model fitting and model selection, respectively. This can improve the performance of EDAs, no matter how simple or complex the learning rule is. We adopt a fast learning method ( $\alpha=1$  and  $\beta=0$ ) in this paper, and an elitism strategy is adopted to maintain a smooth convergence process.

*2.4.2. Probability Model.* In this paper, the normal pdf  $N(\mu_i, \sigma_i)$  for variables  $x_i$  is parameterized by the u of means and covariance  $\sigma$ , which is defined by

$$N(x_{i}, \mu_{i}, \sigma_{i}) = \frac{1}{\sigma_{i} \sqrt{2\pi}} e^{-(x_{i} - \mu_{i})^{2} / 2\sigma_{i}^{2}}.$$
 (7)

The probability distribution  $P(x_1, x_2, ..., x_m)$  of the vector  $(x_1, x_2, ..., x_m)$  of m variables is

$$P(x_1, x_2, \dots x_m) = \prod_{i=1}^{m} \frac{1}{\sigma_i \sqrt{2\pi}} e^{-(x_i - \mu_i)^2 / 2\sigma_i^2}.$$
 (8)

The parameters  $(\mu_i, \sigma_i)$  have been estimated according to the selected best individuals. The estimation of marginal parameters  $(\mu_i, \sigma_i)$  is updated in every iteration.

2.5. Probability Sampling. The probability sampling is used to generate new individuals using the learned probabilistic models. The sampling method depends on the type of probabilistic model and the characteristics of the problem. For normal pdf problem, a conversion is used in order to convert the normal pdf to a standard normal pdf.

Suppose

$$y = \frac{x - \mu}{\sigma}. (9)$$

The normal pdf about *x* is converted to a standard normal pdf about *y*. Consider

$$N(x, \mu, \sigma) \longrightarrow N(y, 0, 1).$$
 (10)

The variable *x* can be calculated by

$$x = \sigma y + \mu. \tag{11}$$

In the probability models, every variable  $(x_1, x_2, ..., x_m)$  is assumed independent of any variable. The mean and variance of variable  $x_i$  are  $\mu_i$  and  $\sigma_i$ ; when  $n \to \infty$ ,

$$y = \frac{\left(\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \mu_i\right)}{\sqrt{\sum_{i=1}^{n} \sigma_i^2} \to N(y, 0, 1)}.$$
 (12)

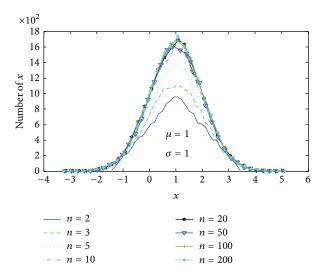


FIGURE 2: Cartogram of sampling data.

If  $x_i$  is the evenly distributed random number of [0, 1],

$$\mu_i = E(x) = \frac{1}{2},$$

$$\sigma_i^2 = V(x) = \frac{1}{12}.$$
(13)

Therefore,

$$y = \frac{\left(\sum_{i=1}^{n} x_i - n/2\right)}{\sqrt{n/12}} \tag{14}$$

when  $n \to \infty$  and  $y \to N(0, 1)$ . We can select an appropriate n to generate a normal pdf for probability sampling. Figure 2 shows the cartogram of sampling data in different n. From the figure, we can see the sampling data following the pdf better with the increasing of n.

2.6. Elitism Strategy. Elitism strategy is an effective strategy to ensure that the best individual is selected as the next generation in EAs, because the best individual may include the information of optimal solution. Therefore, elitism can improve the convergence performance of EAs in many cases [28], and elitism has long been considered an effective method for improving the efficiency of EAs [29]. This is achieved by simply copying the best individual directly to the new generation [30]. However, the number of best individuals selected as the next generation must be handled properly and carefully; otherwise it may lead to premature convergence or cannot improve the efficiency of algorithm. Figure 3 is the process of new population generation. The elitism individuals will be selected as the new generation directly, and the best individuals are used to establish a probability model to generate other individuals of the next generation. Consider

$$Pop(k) = Elitism(BN)_{k-1} \cup Sample(N - BN)_{k-1}, \quad (15)$$

where Elitism(BN) is the operator to copy the best solution of Pop(k-1) to Pop(k) Sample() is the sampling function,

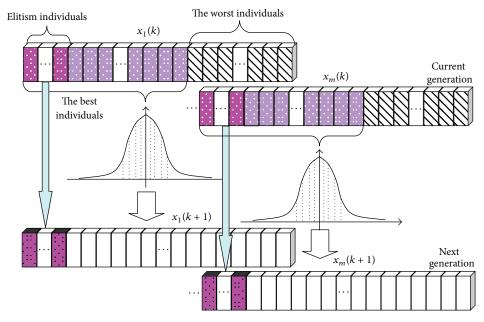


FIGURE 3: Population operation diagram.

N is the population size, and BN is the best selected individuals number.

### 3. Simulation

In the simulation, in order to exhibit the performance of FEGEDA, we carry out several different simulations, including one-dimensional benchmark, two-dimensional benchmarks, and higher dimensional benchmarks. Moreover, we compare the FEGEDA with other EDAs and other kinds of optimization algorithms.

3.1. One-Dimensional Benchmark. One-dimensional problem is easy for FEGEDA. In order to visualize the information of optimization process and models learning process during the evolution clearly, we carry out a one-dimensional benchmark optimization simulation:

$$f_0(x) = \sin(x) + \sin(\frac{10}{3}x) + \log(x) - 0.84x + 3,$$
 (16)

where  $f_0$  is a multimodal [31],  $x \in [2.7, 7.5]$ , with several local minimum value, and the global minimum value 1.6013 at x = 5.19978.

The best individuals number *BN* selected to build the probability model is a very important parameter for FEGEDA. The elitism strategy is a very important strategy to maintain a smooth optimization process in this paper. Therefore, in order to visualize the performance of corresponding part, we use different BN to testify the effect of outstanding individuals No. to the algorithm performance, and the elitism strategy is optional to testify the effect of the elitism strategy to the convergent performance of the algorithm. Many literatures [32–34] have proved that EDAs are convergent under certain conditions. From Figure 4, we can see that the optimization processes are unstable due to the

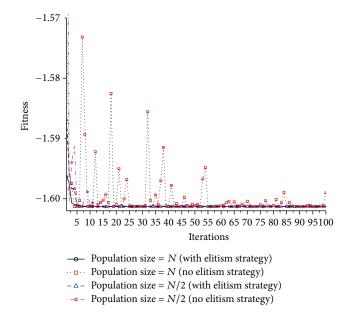


FIGURE 4: The optimum solutions of each iteration under different conditions.

use of fast learning rule when the algorithm is without elitism strategy no matter what BN is. The elitism strategy can make the convergent process smooth and improve the convergent performance too.

In Figure 5, the individuals' distribution and probability models of some iteration are exhibited. The individuals' distributions of iterations 1, 10, and 100 are shown in the Figure 5. The individuals spread over the optimized function at initial iteration, and then the individuals will focus on the area of optimum solution with the iterations going on.

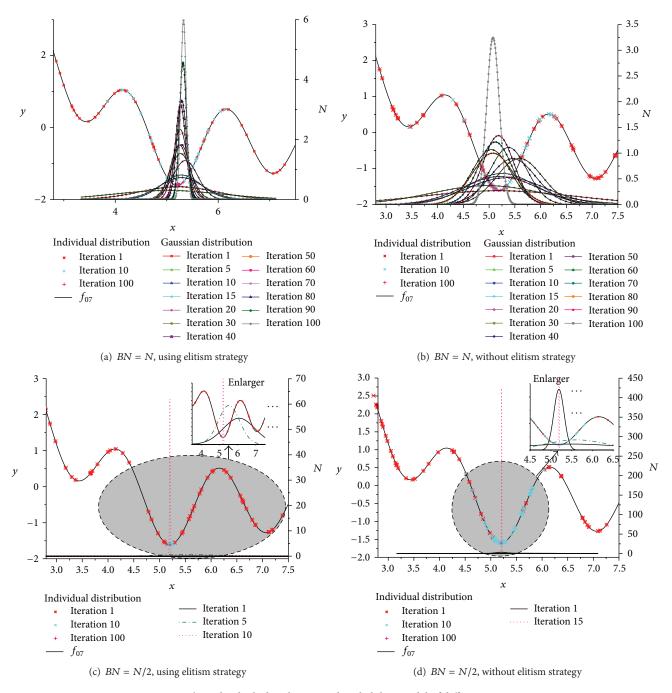


FIGURE 5: The individuals distribution and probability model of different iterations.

Therefore, the diagram of pdf is flat at the beginning. The parameter  $\mu$  of pdf is smaller and smaller with the increase of iteration and focus on the global optimum solution. The probability models learning process is shown in Figure 5. The elitism strategy is a very important part of the algorithm. Form the exhibition of probability models learning process in Figure 5, we can see that the probability model learning process of solution is smooth when adopting elitism strategy; otherwise it is unstable.

The best selected individuals number is also an important parameter. The convergent speed is faster when the best selected individuals number BN is N/2. However, if it is too small, it will lead to premature.

Figure 6 is the statistics information of population of some iteration. Form Figure 6, we can see the population distribution when BN = N using elitism strategy (Figure 6(a)) or without elitism strategy (Figure 6(b)), and BN = N/2 using elitism strategy (Figure 6(c)) or without elitism strategy (Figure 6(d)). According to Figure 6, the distribution of population is stable when using elitism strategy; otherwise it is fluctuant regardless of BN = N or BN = N/2. A small BN can make the individuals focus on a certain area quickly.

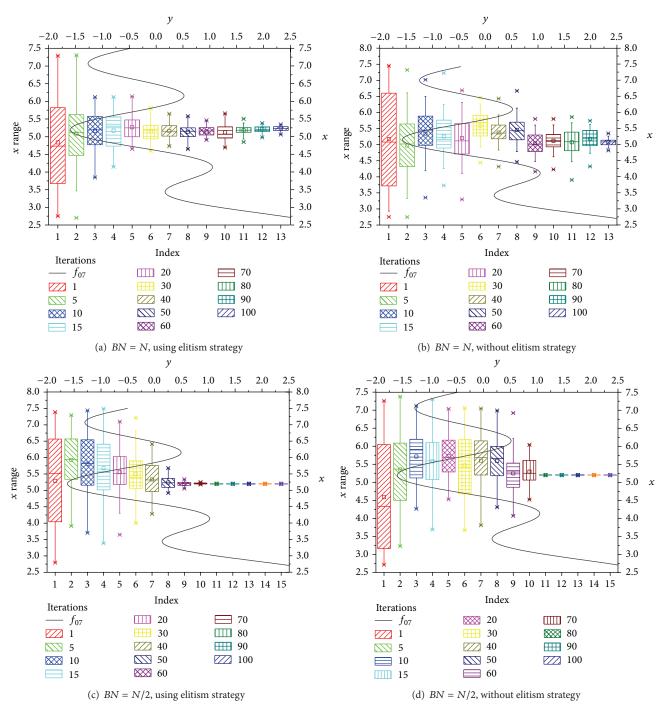


FIGURE 6: The boxplot of population for different iterations.

*3.2. Two-Dimensional Problems.* In order to testify the optimization capability of FEGEDA further, three two-dimensional complex functions are considered:

$$f_1(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001 * (x^2 + y^2)^2)^2},$$

$$f_2(x, y) = \left(\frac{3}{0.05 + (x^2 + y^2)^2}\right)^2 + x^2 + y^2$$

$$f_3(x,y) = -(x^2 + y^2)^{0.25} \left(\sin^2 50 * (x^2 + y^2)^{0.1} + 0.1\right),$$
(17)

where  $x, y \in [-5.12, 5.12]$ .  $f_1$  has infinite maximum value, and the global maximum value 1 is point (0,0). A circuit ridge surrounds the global maximum value. Hence, it is easy to fall into local maximum, which can be used to test the global searching capability of the algorithm.  $f_2$  is a local peak function, and the maximum value is 3600 at point (0,0). This function can be used in determining the local

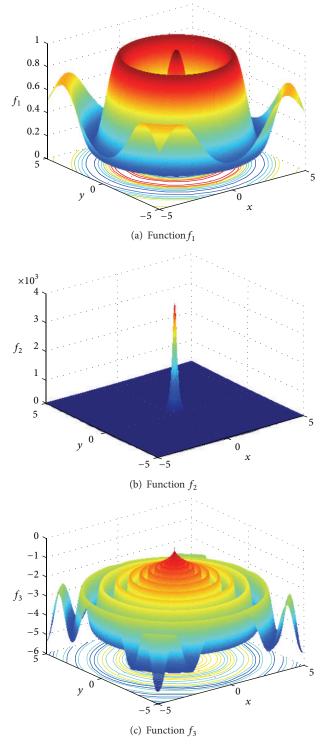


FIGURE 7: Function diagrams of  $f_1$ ,  $f_2$ , and  $f_3$ .

searching capability of the algorithm. The  $f_3$  function is a complicated function with a vibrating circuit ridge outside the global maximum value 0. This function can verify the global and local optimization capability of the algorithm simultaneously. Figure 7 shows the functions  $f_1$ ,  $f_2$ , and  $f_3$  correspondingly. We compare the optimization result with three other algorithms [35].

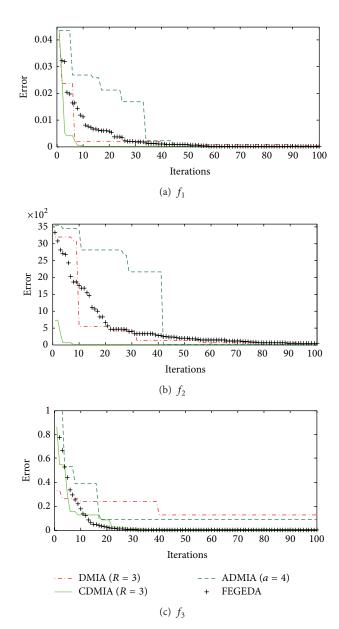


FIGURE 8: Comparisons of convergent results.

The population size N is set to 50, the maximum iteration is set to 100, and BN is set to N/2 in order to have comparison under the same conditions. From Figure 8, we can see that the FEGEDA can get the optimum solution faster. It has similar optimization capability of CDMIA, which has preferable performance for the three benchmarks.

3.3. Higher Dimensional Problems. The advantage of FEGEDA is the capability of higher dimensional problems solution. Some typical benchmarks are considered, including Quadric, Rosenbrock, Ackley, Griewank, Rastrigrin, and Schaffer's  $f_7$  function [21], which are shown in Table 1. In addition, they are configured with a dimension n=10. In order to compare with other EDAs under the same

TABLE 1: High dimensional benchmarks.

Quadric 
$$n = 10$$
 
$$f_4(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)^2, \quad -100 \le x_i \le 100$$
 Rosenbrock  $n = 10$  
$$f_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2\right)^2 + \left(1 - x_1\right)^2\right], \quad -50 \le x_i \le 50$$
 Ackley  $n = 10$  
$$f_6(x) = -20 \exp\left(-0.2 \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}\right) - \exp\left(\sqrt{\frac{\sum_{i=1}^n \cos{(2\pi x_i)}}{n}}\right) + 20 + e, \quad -30 \le x_i \le 30$$
 Griewank  $n = 10$  
$$f_7(x) = \frac{1}{4000} \sum_{i=1}^n (x_i)^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad -300 \le x_i \le 300$$
 Rastrigrin  $n = 10$  
$$f_8(x) = \sum_{i=1}^n \left[x_i^2 - A\cos{(2\pi x_i)} + A\right], \quad -5.12 \le x_i \le 5.12$$
 Schaffer's  $f_7$   $n = 10$  
$$f_9(x) = \sum_{i=1}^n \left(x_i^2 + x_{i+1}^2\right)^{0.25} \times \left\{\sin\left[50 \times \left(x_i^2 + x_{i+1}^2\right)^{0.1}\right] + 1.0\right\}, \quad -100 \le x_i \le 100$$

conditions, the population size N of FEGEDA is 300 and the maximum iteration is 100.

The algorithm is testified under different BN (from N to N/20). The convergent results under different BN are shown in Figure 9. Form Figure 9, we can see that the optimization process is slow when BN = N. With the decrease of BN, the convergent speed is faster. However, the increase of convergent speed is limited. If the BN is too small, the optimization will trap into local minimum easily.

We have a comparison of FEGEDA with other EDAs in [21]. Figure 10 is the comparison diagram. From Figure 10, we can see that FEGEDA is superior to standard EDA and other improved ones for the six benchmarks. For Ackley function, the performance of FEGEDA is the same as EDA. For Rosenbrock function, the initial fitness of FEGEDA is lower than other EDAs. Therefore, we put an enlarger diagram of corresponding area.

# 4. PID Controller Optimization

PID is the most used controller in the permanent magnet synchronous motors (PMSM) control. However, PID controller has poor performance in PMSM control due to the inappropriate parameters. During the past decades, great attention has been paid to the stochastic approach, which is potential in dealing with the problem [36, 37]. In this paper, we adopt FEGEDA to optimize the PID controller of PMSM.

4.1. Mathematic Model of PMSM. The mathematical model of PMSM in a d, q two-phase rotating coordinate system is shown below [38]. The voltage equation is

$$u_q = R_s i_q + L_q \dot{i}_q + \omega_e L_d i_d + \omega_e \psi_f,$$

$$u_d = R_s i_d + L_d \dot{i}_d - \omega_e L_q i_q,$$
(18)

where  $u_d$  and  $u_q$  represent the stator winding shaft in a straight axis and the quadrature voltage, respectively;  $i_d$  and  $i_q$  are the direct-axis current and quadrature-axis current, respectively;  $R_s$  is the stator phase resistance;  $L_d$  is the straight axis inductance;  $L_q$  is the quadrature axis inductance;

 $\psi_f$  is the permanent-magnet fundamental excitation magnetic field and stator winding of the magnetic chain;  $w_e$  is the electric angular speed of rotor.

The magnetic linkage equation can be expressed as follows:

$$\psi_d = L_d i_d + \psi_f,$$
 
$$\psi_q = L_q i_q,$$
 (19)

where  $\psi_d$  and  $\psi_q$  represent the syntheses of the magnetic fields in space-direct and quadrature-axis stator winding of the magnetic chain, respectively.

The electromagnetic torque of PMSM in the d, q coordinate is

$$T_e = p_n \left( \psi_f i_q - \left( L_d - L_q \right) i_p i_d \right), \tag{20}$$

where  $p_n$  is number of pole pairs.

According to the motion equation of motor,

$$Jp\dot{\Omega}_{r} = T_{e} - T_{l} - B\Omega_{r},$$

$$\Omega_{r} = \frac{\omega_{e}}{p_{n}},$$
(21)

where  $\Omega_r$  is mechanical angular speed of rotor, B is the viscous friction coefficient, J is the total moment inertia of rotor and load, and  $T_l$  is the load torque.

Thus, the state equation can be derived from the above equations as follows:

$$\dot{i}_{q} = \frac{1}{L_{q}} \left( u_{q} - R_{s}i_{q} - L_{d}i_{d}w_{e} - \psi_{f}w_{e} \right),$$

$$\dot{i}_{d} = \frac{u_{d}}{L_{d}} \left( u_{d} - R_{s}i_{d} - w_{e}L_{q}i_{q} \right),$$

$$\dot{w}_{e} = \frac{1.5 p_{n}^{2} \left( \psi_{f}i_{q} + \left( L_{d} - L_{q} \right) i_{d}i_{q} \right) - p_{n}T_{m} - Bw_{e}}{I}.$$
(22)

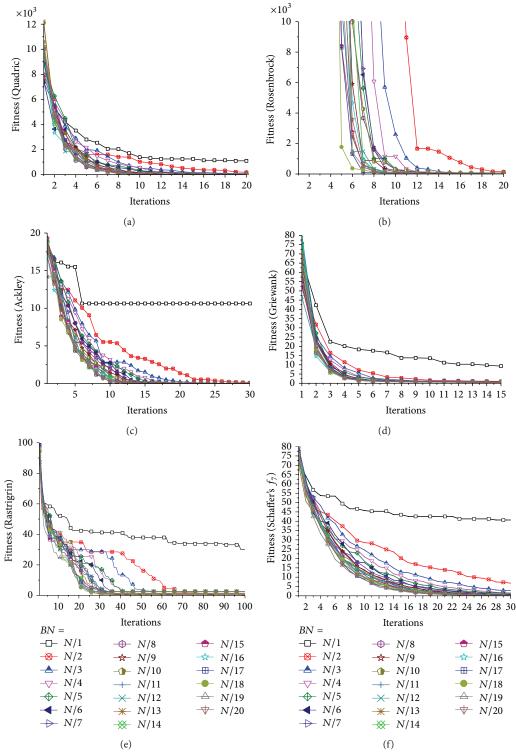


FIGURE 9: The convergent results under different BN.

In the VC system of PMSM,  $i_d=0$ . Therefore, the state space equation (22) is described as

$$\dot{i}_{q} = \frac{1}{L_{q}} \left( u_{q} - R_{s} i_{q} - \psi_{f} w_{e} \right), 
\dot{w}_{e} = \frac{1.5 p_{n}^{2} \psi_{f} i_{q} - p_{n} T_{m} - B w_{e}}{J}.$$
(23)

4.2. PID Controller. The continuous form of a PID controller, with input e and output u, is shown as follows:

$$u(t) = K_p e(t) + K_i \int e(t) + K_d \dot{e}(t),$$
 (24)

where  $K_p$  is the proportional gain,  $K_i$  is the integral gains, and  $K_d$  is the derivative gains.

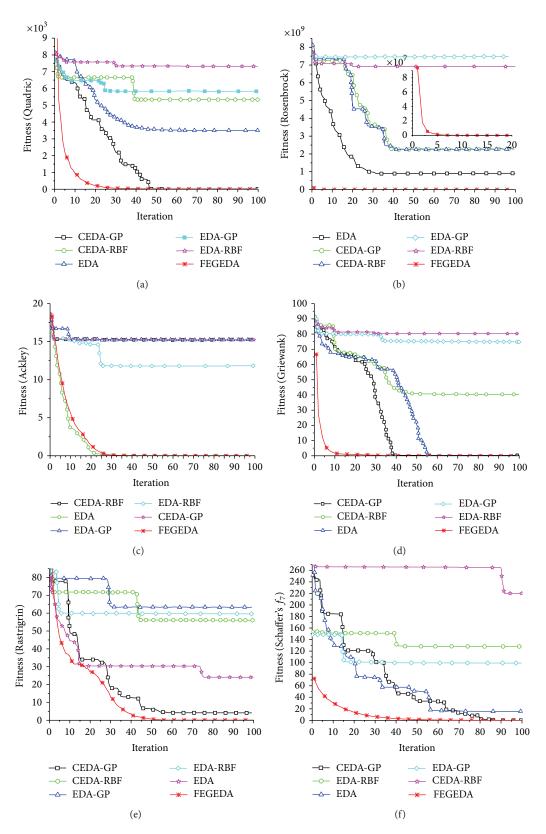


FIGURE 10: Comparisons of convergent results.

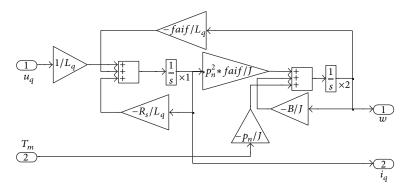


FIGURE 11: MATLAB/Simulink model of PMSM.

There are two types of discrete PID by discretization of continuous PID. The position-type discrete PID is described as

$$u(k) = K_{p}e(k) + K_{i} \sum_{j=0}^{k} T_{s}e(k) + \frac{K_{d}}{T_{s}} (e(k) - e(k-1)),$$
(25)

where u(k) is the controller output, e(k) is the error. In practical system control, the integral part is not flexible. Therefore, another velocity-type discrete PID is described as

 $\Delta u(k)$ 

$$= K_p \Delta e(k) + K_i T_s e(k) + \frac{K_d}{T_s} (\Delta e(k) - \Delta e(k-1)), \quad (26)$$

$$\Delta e(k) = e(k) - e(k-1),$$

where  $T_s$  is the sample time. For velocity-type PID, we do not need to calculate the integral part, and the controller output is the increment of PID. Therefore, it is often used in practical system control.

Aggregation function is a conventional method, which can convert a multiobjective problem to a single-objective problem. Consider

$$fitness = \sum_{i=1}^{n} w_i f_i, \tag{27}$$

where fitness is the summation of fitness,  $w_i$  is the weight of ith objective, and  $f_i$  is the fitness value of ith objective.

In the optimization process, the objective is to evaluate the performance of PIDs. Thus, for PID, the fitness function is written as [39]

$$f_1 = \int_0^\infty |e(t)| dt$$

$$f_2 = \int_0^\infty u^2(t) dt$$

$$f_3 = t_r,$$
(28)

where e(t) is the system error, u(t) is the control output, and  $t_r$  is the rising time.

To avoid overshoot, a penalty value is adopted in the fitness function. That is, once overshoot occurs, the value of overshoot is added to the fitness function. Hence, the penalty function is written as

$$f_4 = \begin{cases} \int_0^\infty (y(t) - y(t-1)) dt & \text{if } e(t) < 0\\ 0 & \text{if } e(t) \ge 0, \end{cases}$$
 (29)

where y(t) is the control output.

Making use of the aggression function, the fitness function is constructed as follows:

$$f = w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4, \tag{30}$$

where  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$  are the weight coefficients, and  $w_4 \gg w_1$ .

4.3. PID Controller Optimization Based on FEGEDA. According to state space equation (6), we can build the state space model of PMSM in MATLAB/Simulink as Figure 11. The parameters of PMSM are that  $R_s$  is 0.9664,  $L_q$  is 0.00621,  $P_n$  is 4, J is 0.00033, B is 0.0001619, and  $\psi_f$  is 0.09382 according to motor.

The component of PMSM is encapsulated into a module. A speed controller added to the speed closed loop. Figure 12 is the diagram of PMSM control system. The "simouterror," "simoutui," and "simout" units are used to record the simulation data for optimization.

In order to testify the algorithm, GA and traditional PID are selected to compare against FEGEDA.  $w_1, w_2, w_3$ , and  $w_4$  of  $f_i$  are set according to the requirement of control system.  $w_1$  is corresponding to the control variable of error,  $w_2$  is a weight coefficient of controlled variable,  $w_3$  is for the control variable of rising time, and  $w_4$  is the penalty of overshoot. If we want a system without overshoot and have a small rising time,  $w_1$ ,  $w_3$ , and  $w_4$  will be set bigger, and  $w_2$  is smaller. If the controlled variable is limited,  $w_2$  will be set bigger. Therefore, these parameters can be set according to the practical requirement. In the test,  $w_1$  is 1,  $w_2$  is 0.1,  $w_3$  is 2, and  $w_4$  is 200. The parameters of GA are that the population size is 30, crossover probability is 0.9, and mutation probability is adaptive to individual fitness. The variable domain of  $K_p$  is [0,20] and  $K_i$  and  $K_d$  are [0,1]. The iteration number is

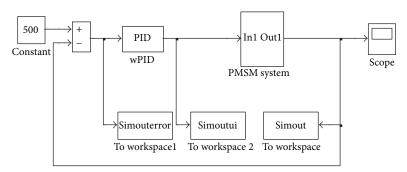


FIGURE 12: The diagram of PMSM control system.

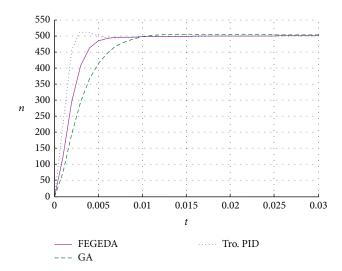


FIGURE 13: The system response of PMSM with different PIDs.

50. The optimal results are shown in Figure 13. Although the optimal result of traditional PID has shorter response time, the overshoot is bigger. The optimal result of GA has no overshoot, but the system response is slower. Therefore, the performance of FEGEDA is promising.

#### 5. Conclusions

We studied the estimation of distribution algorithm in this paper and proposed a fast elitism Gaussian EDA for continuous optimization. Every dimension of individuals is represented by means and standard deviations of Gaussian distribution. These parameters are estimated using maximum likelihood technique by fast learning rule. Then the new population is generated by sampling and elitism strategy. The elitism strategy improves the convergent performance of the algorithm. In the one-dimensional test, we exhibit the optimization process and probability models learning process clearly. In the two-dimensional and higher dimensional problems, we compare the FEGEDA with danger immune algorithm and other EDAs, and the FEGEDA exhibits a good performance. Although the performance of FEGEDA is promising, further studies are still recommended, for

instance, how to increase the diversity of population under fast convergence rate.

### **Conflict of Interests**

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

# Acknowledgment

This work is supported by the National Natural Science Foundation of China (under Grant 61174044).

#### References

- [1] B. A. H. Al-Sarray and R. A. D. Al-Dabbagh, "Variants of hybrid genetic algorithms for optimizing likelihood ARMA model function and many of problems," in *Evolutionary Algorithms*, pp. 219–246, InTech, Winchester, UK, 2011.
- [2] S. Das, S. Maity, B. Qu, and P. N. Suganthan, "Real-parameter evolutionary multimodal optimization—a survey of the state-of-the-art," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 71–88, 2011.
- [3] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [4] S. Forrest, L. Allen, A. S. Perelson, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, pp. 202–212, May 1994.
- [5] X. Cao, H. Qiao, and Y. Xu, "Negative selection based immune optimization," *Advances in Engineering Software*, vol. 38, no. 10, pp. 649–656, 2007.
- [6] G. Wu, W. Pedrycz, M. Ma, D. Qiu, H. Li, and J. Liu, "A particle swarm optimization variant with an inner variable learning strategy," *The Scientific World Journal*, vol. 2014, Article ID 713490, 15 pages, 2014.
- [7] R. Liu, C. Ma, W. Ma, and Y. Li, "A multipopulation PSO based memetic algorithm for permutation flow shop scheduling," *The Scientific World Journal*, vol. 2013, Article ID 387194, 11 pages, 2013.
- [8] N. Belgasmi, L. Ben Said, and K. Ghédira, "Evolutionary multiobjective optimization of the multi-location transshipment problem," *Operational Research*, vol. 8, no. 2, pp. 167–183, 2008.

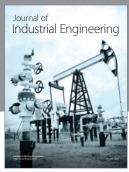
- [9] M. Pelikan, D. E. Goldberg, and F. G. Lobo, "A survey of optimization by building and using probabilistic models," *Computational Optimization and Applications*, vol. 21, no. 1, pp. 5–20, 2002.
- [10] H. Miuhlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters," in Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, pp. 178–187, London, UK, 1996.
- [11] L. Li, H. Chen, C. Liu et al., "A robust hybrid approach based on estimation of distribution algorithm and support vector machine for hunting candidate disease genes," *The Scientific World Journal*, vol. 2013, Article ID 393570, 7 pages, 2013.
- [12] U. Aickelin, E. K. Burke, and J. Li, "An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering," *Journal of the Operational Research Society*, vol. 58, no. 12, pp. 1574–1585, 2007.
- [13] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111–128, 2011.
- [14] S. Shahraki and M. R. A. Tutunchy, "Continuous Gaussian estimation of distribution algorithm," in *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*, pp. 211–218, Springer, Berlin, Germany, 2013.
- [15] A. Zhou, B. Qu, H. Li, S. Zhao, P. N. Suganthan, and Q. Zhangd, "Multiobjective evolutionary algorithms: a survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [16] S. Baluja, "Population-based incremental learning. A method for integrating genetic search based function optimization and competitive learning," Technical Report CMU-CS-94-163, DTIC Document, Pittsburgh, Pa, USA, 1994.
- [17] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 287–297, 1999.
- [18] S. Rudlof and M. Köppen, "Stochastic hill climbing with learning by vectors of normal distributions," in *Proceedings of* the 1st On-line Workshop on Soft Computing, pp. 60–70, Nagoya, Japan, 1996.
- [19] M. E. L. Sebag and A. Ducoulombier, "Extending population-based incremental learning to continuous search spaces," in *Proceedings of the 5th Parallel Problem Solving from Nature (PPSN '89)*, pp. 418–427, Amsterdam, The Netherlands, 1998.
- [20] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Regularized continuous estimation of distribution algorithms," *Applied Soft Computing*, vol. 13, no. 5, pp. 2412–2432, 2013.
- [21] N. Luo, F. Qian, L. Zhao, and W. Zhong, "Gaussian process assisted coevolutionary estimation of distribution algorithm for computationally expensive problems," *Journal of Central South University of Technology*, vol. 19, no. 2, pp. 443–452, 2012.
- [22] H. Li, Y. Hong, and S. Kwong, "Subspace estimation of distribution algorithms: to perturb part of all variables in estimation of distribution algorithms," *Applied Soft Computing Journal*, vol. 11, no. 3, pp. 2974–2989, 2011.
- [23] M. Nakao, T. Hiroyasu, M. Miki, H. Yokouchi, and M. Yoshimi, "Real-coded estimation of distribution algorithm by using probabilistic models with multiple learning rates," in *Proceedings of the 11th International Conference on Computational Science (ICCS '11)*, vol. 4, pp. 1244–1251, June 2011.
- [24] P. A. N. Bosman and D. Thierens, "Numerical optimization with real-valued estimation-of-distribution algorithms," *Studies in Computational Intelligence*, vol. 33, pp. 91–120, 2007.

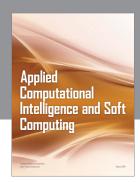
- [25] L. Martí, J. Garca, A. Berlanga, C. A. Coello Coello, and J. M. Molina, "MB-GNG: addressing drawbacks in multi-objective optimization estimation of distribution algorithms," *Operations Research Letters*, vol. 39, no. 2, pp. 150–154, 2011.
- [26] P. A. Bosman and J. Grahl, "Matching inductive search bias and problem structure in continuous Estimation-of-Distribution Algorithms," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1246–1264, 2008.
- [27] M. Gallagher, M. Frean, and T. Downs, "Real-valued evolutionary optimization using a flexible probability density estimator," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 840–846, Orlando, Fla, USA, 1999.
- [28] C. W. Ahn and R. S. Ramakrishna, "Elitism-based compact genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 367–385, 2003.
- [29] R. C. Purshouse and P. J. Fleming, "Why use elitism and sharing in a multi-objective genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 520–527, New York, NY, USA, 2002.
- [30] I. J. Leno, S. S. Sankar, M. V. Raj, and S. G. Ponnambalam, "An elitist strategy genetic algorithm for integrated layout design," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9–12, pp. 1573–1589, 2013.
- [31] L. Lasdon, A. Duarte, F. Glover, M. Laguna, and R. Martí, "Adaptive memory programming for constrained global optimization," *Computers and Operations Research*, vol. 37, no. 8, pp. 1500–1509, 2010.
- [32] H. Muhlenbein, "Convergence theorems of estimation of distribution algorithms," in *Markov Networks in Evolutionary Computation*, pp. 91–108, Springer, Berlin, Germany, 2012.
- [33] H. Miuhlenbein, J. U. R. Bendisch, and H. Voigt, "From recombination of genes to the estimation of distributions II. Continuous parameters," in *Proceedings of the 4th Parallel Problem Solving from Nature (PPSN '96)*, pp. 188–197, Springer, Berlin, Germany, 1996.
- [34] Q. Zhang and H. Mühlenbein, "On the convergence of a class of estimation of distribution algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 127–136, 2004.
- [35] Q. Xu, S. Wang, L. Zhang, and Y. Liang, "A novel chaos danger model immune algorithm," Communications in Nonlinear Science and Numerical Simulation, vol. 18, no. 11, pp. 3046–3060, 2013
- [36] L. Huang, N. Wang, and J. Zhao, "Multiobjective optimization for controller design," *Acta Automatica Sinica*, vol. 34, no. 4, pp. 472–477, 2008.
- [37] W. Wojsznis, A. Mehta, P. Wojsznis, D. Thiele, and T. Blevins, "Multi-objective optimization for model predictive control," ISA Transactions, vol. 46, no. 3, pp. 351–361, 2007.
- [38] G. Chen, J. Kang, and J. Zhao, "Numeric analysis and simulation of space vector pulse width modulation," *Advances in Engineering Software*, vol. 65, pp. 60–65, 2013.
- [39] J. Zhang, J. Zhuang, H. Du, and S. Wang, "Self-organizing genetic algorithm based tuning of PID controllers," *Information Sciences*, vol. 179, no. 7, pp. 1007–1018, 2009.

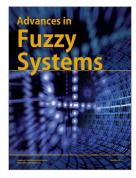
















Submit your manuscripts at http://www.hindawi.com

