



# The Open Cybernetics & Systemics Journal

Content list available at: [www.benthamopen.com/TOCSJ/](http://www.benthamopen.com/TOCSJ/)

DOI: 10.2174/1874110X01610010020



## Adaptive Learning Rate Elitism Estimation of Distribution Algorithm Combining Chaos Perturbation for Large Scale Optimization

Qingyang Xu<sup>\*</sup>, Chengjin Zhang, Jie Sun and Li Zhang

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

**Abstract:** Estimation of distribution algorithm (EDA) is a kind of EAs, which is based on the technique of probabilistic model and sampling. Large scale optimization problems are a challenge for the conventional EAs. This paper presents an adaptive learning rate elitism EDA combining chaos perturbation (ALREEDA) to improve the performance of traditional EDA to solve high dimensional optimization problems. The famous elitism strategy is introduced to maintain a good convergent performance of this algorithm. The learning rate of  $\sigma$  (a parameter of probabilistic model) is adaptive in the optimization to enhance the algorithm's global and local search ability, and the chaos perturbation strategy is used to improve the algorithm's local search ability. Some simulation experiments are conducted to verify the performance of ALREEDA by seven benchmarks of CEC'08 large scale optimization with dimensions 100, 500 and 1000. The results of ALREEDA are promising on majority of the testing problems, and it is comparable with other EDAs and some other improved EAs.

**Keywords:** Adaptive learning rate, Chaos perturbation, Elitism, Estimation of distribution algorithm, Gaussian distribution.

### 1. INTRODUCTION

Large-scale continuous optimization problems using heuristic algorithms have been one of the most interesting trends in the last few years. Many real world problems may be modeled as large scale problems in continuous domains, such as large combinational problem [1], computer network management [2], industrial engineering [3], *etc.* Although, evolutionary algorithms (EAs) have been successfully applied to various engineering and scientific problems for the last two decades [4, 5], the classical EAs often lose their efficacy and advantages when applied to large scale and complex problems [6]. Numerous improved heuristic algorithms have been developed to enhance the large scale optimization ability of algorithms, such as improved particle swarm optimization (PSO), differential evolution (DE), ant colony optimization (ACO) [7, 8], *etc.* The cooperative coevolution is an effective approach to solve large and complex problems [6, 9]. A multilevel cooperative coevolution (MLCC) with a decomposer pool was proposed by Yang and Tang [10], to divide the objective vector problem into several subcomponents.

In recent years, other type of algorithm called Estimation of Distribution Algorithm (EDA) has attracted a lot of attention. It was proposed by Miuhlenbein and Paaß [11], and emerged as a generalization of EAs, for overcoming intrinsic disadvantages of EAs, like building blocks broken, poor performance in high dimensional problems and the difficulty of modeling the solution distribution. Compared with blocks building in EAs, EDA has some attractive characteristics. It does not use the recombination or mutation operators.

Instead, they extract the global statistical information from the superiority individual and build the probability model of solution distribution. It is the main advantage of EDA over EAs that the explanatory and transparency of the probabilistic model guide the search process [12, 13]. The new solutions come from the sampling of established probability model which approximates the distribution of promising solutions [14]. Such reproduction procedure allows EDA to search for the global optimal solutions effectively. Additionally, the priori information about the problem can

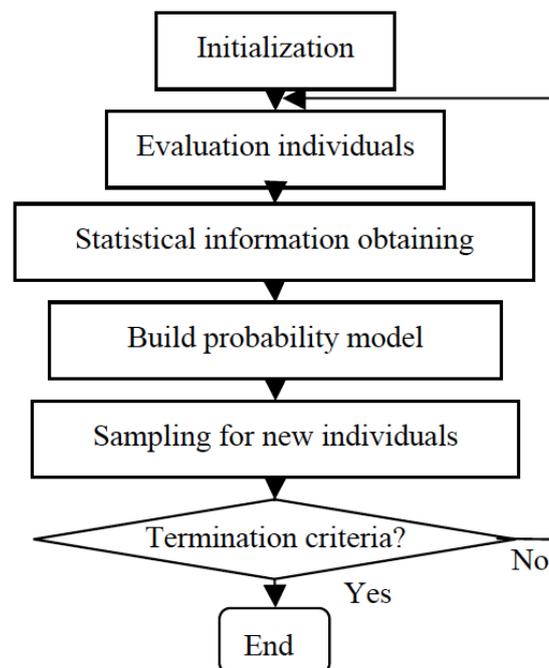
<sup>\*</sup> Address correspondence to this author at the School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai, China; E-mail: [xuqy1981@163.com](mailto:xuqy1981@163.com)

captured by the probability model [15]. There are many researches about the EDAs. Karshenas and Santana [16] adopt regularized method to improve the conventional EDA performance, which used some benchmarks with 100 dimensions for testing. Valdez and Hernández [17] adopt Gaussian model to approximate the Boltzmann distribution, which can analyze the minimization of the Kullback-Leibler divergence instead of computing the mean and variance of Gaussian model, and the test suite is up to 50 dimensions. Ahn [4] combined PSO with EDA to improve the performance of EDA. There are also many other improved EDAs to enhance the performance of EDA in various domains [14, 18 - 26]. However, the dimension of the problems is up to 100. The EDAs rarely seen are applied to solve large scale optimization problem. Wang and Li proposed a robust univariate EDA (LSEDA-gl) [27] for large scalar global optimization, which made use of mixed Gaussian and Levy probability distribution for sampling. An Estimation of Distribution and Differential Evolution Cooperation (ED-DE) algorithm [28] was proposed to form a new cooperative optimizer, and used for large scale economic load dispatch optimization of power systems. Ata and Bootkrajang [29] employed multiple random projections of the fit individual, and carried out model estimation and individuals sampling in lower dimensional spaces. It is more efficient and reliable than working in the original high dimensional space.

In some EDAs, many probability models and mixtures of pdfs are involved [30]. However, the probability models cannot reflect the problem completely, especially for the increases of number of variables and the number of mixture components, the optimization results become unreliable [31]. Additionally, the computational cost is huge when considering all the possible (in) dependencies among the variables [32]. Therefore, in this paper we only adopt univariate Gaussian model to approximate the solution distribution. In the Gaussian model, some parameters are learnable. In this paper, we propose an adaptive learning rate elitism EDA combining chaos perturbation search strategy for large scale optimization problem. The learning rate of Gaussian parameter is adaptive in the optimization process. We also adopt an elitism strategy to enhance the convergent performance of the algorithm, which is a popular strategy in EAs. In order to improve the local search ability, a chaos perturbation operator is designed. The local search operator enhances the diversity of population in the iteration.

## 2. ESTIMATION OF DISTRIBUTION ALGORITHM

Estimation of distribution algorithm is a series of EAs based on probability theory, which makes use of estimation and sampling technology to approximate solutions distribution and generate new solutions. The Fig. (1) is the flowchart of EDA.



**Fig. (1).** Flowchart of EDA.

### 2.1. The Probability Model Build and Updating Mechanism

The most important and crucial step of EDAs is how to build the probabilistic model  $P(x)$  to express the promising solutions. In EDAs for global continuous optimization problem, the Gaussian distribution is a common one. Some other complex models, like Gaussian mixture, histogram *etc.*, are also used [33]. It must be noted that, different dependency relations can appear between variables for large-scale optimization problems. Considering all the dependencies needs a complex probability model, which is hard to realize. Therefore, we specifically focus on the use of univariate probability model to reduce the amount of calculation. Even though the dependency between variables is very important for system modeling, the optimum solution is the most important issue for optimization problem. Hence, every variable is assumed independent of any variable in the probability model. The Gaussian model is used to model and estimate the distribution of promising solutions in every dimension of the problem.

In order to construct a Gaussian 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 [31]. In the algorithm assuming full independence, every variable is assumed independent of any variable. That is, the probability distribution  $P(x_1, x_2, \dots, x_D)$  of the vector  $(x_1, x_2, \dots, x_D)$  of  $m$  variables is assumed to consist of a product of the distributions of individual variables:

$$P^k(x_1, x_2, \dots, x_D) = \prod_{i=1}^D N(x_i | \mu_i^k, \sigma_i^k) \quad (1)$$

where  $\mu_i^k$  is the mean and  $\sigma_i^k$  is the standard deviation of  $k$ -th generation and  $i$ -th variable.  $D$  is the dimension size. This is very suitable for calculation. Different from the discrete EDAs, the number of parameters to be estimated does not grow exponentially with  $D$ .

The pdf  $N(x_i | \mu_i^k, \sigma_i^k)$  for variables  $x_i$  is parameterized by the mean  $\mu_i^k$  and the standard deviation, which is defined by

$$N(x_i^k | \mu_i^k, \sigma_i^k) = \frac{1}{\sigma_i^k \sqrt{2\pi}} e^{-\frac{(x_i - \mu_i^k)^2}{2\sigma_i^{k2}}} \quad (2)$$

Therefore, the probability distribution  $P(x_1, x_2, \dots, x_D)$  of the vector  $(x_1, x_2, \dots, x_D)$  of  $m$  variables is

$$P(x_1, x_2, \dots, x_D) = \prod_{i=1}^m \frac{1}{\sigma_i^k \sqrt{2\pi}} e^{-\frac{(x_i - \mu_i^k)^2}{2\sigma_i^{k2}}} \quad (3)$$

The parameters  $(\mu_i^k, \sigma_i^k)$  can be estimated according to the selected best individuals. The parameters  $(\mu_i, \sigma_i)$  can be updated every iteration.

The mean and standard deviation parameters of promising population can be computed adaptively by the maximum likelihood technique according to the selected promising solutions.

$$\mu_i(k) = \frac{1}{NB} \sum_{n=1}^{NB} x_i^n(k) \quad (4)$$

$$\sigma_i^2(k) = \frac{1}{NB} \sum_{n=1}^{NB} (x_i^n(k) - \mu_i(k))(x_i^n(k) - \mu_i(k))^T \quad (5)$$

$\mu_i(k)$  is the mean of  $i$ -th variable in  $k$ -th iteration,  $NB$  is the selected individuals size.  $\sigma_i^2(k)$  is the covariance of  $i$ -th variable in  $k$ -th iteration.

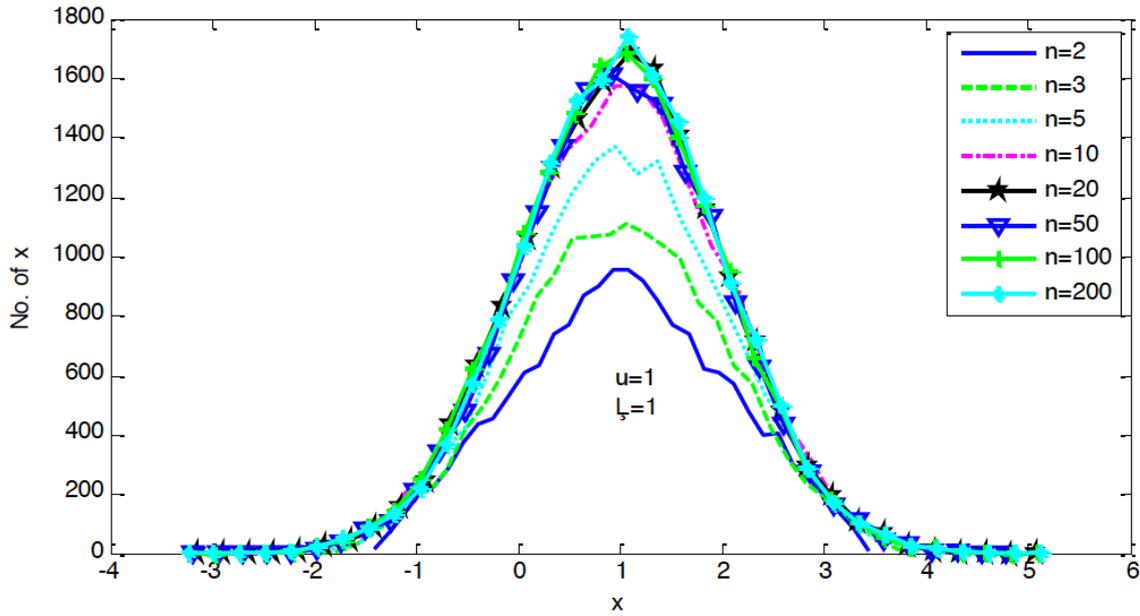


Fig. (2). Cartogram of sampling data.

### 2.2. Probabilistic Sampling

The probability sampling is used to generate new individuals using the learned probabilistic models instead of crossover or mutation operators. The sampling method depends on the type of probabilistic model and the characteristics of the problem. For normal pdf problem, a conversion can be used in order to convert the normal pdf to a standard normal pdf.

Supposing,

$$y = \frac{x - \mu}{\sigma} \tag{6}$$

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

$$N(x | \mu, \sigma) \rightarrow N(y | 0, 1) \tag{7}$$

The variable  $x$  can be calculated by

$$x = \sigma y + \mu \tag{8}$$

In the probability models, every variable  $(x_1, x_2, \dots, x_m)$  is assumed independent of any variable. The mean and standard deviation of variable  $x_i$  is  $\mu_i$  and  $\sigma_i$ , when  $n \rightarrow \infty$ ,

$$y = \left( \sum_{i=1}^n x_i - \sum_{i=1}^n \mu_i \right) / \sqrt{\sum_{i=1}^n \sigma_i^2} \rightarrow N(y, 0,1) \tag{9}$$

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

$$\mu_i = E(x) = 1/2 \tag{10}$$

$$\sigma_i^2 = V(x) = 1/12 \tag{11}$$

Therefore,

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

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

### 3. ADAPTIVE LEARNING RATE ELITISM GAUSSIAN EDA

In this section, we will introduce some special parts in the adaptive learning rate elitism Gaussian EDA, including the adaptive learning rate, chaos perturbation operator and elitism strategy, which are different from standard EDA.

#### 3.1. Adaptive Learning Rate

The parameters of Gaussian model are always learnable in the process of optimization. Some iterative learning approaches are used in some literatures [34 - 37], and we can conclude as follows.

$$\mu_i(k) = \alpha\mu_i(k) + \beta\mu_i(k-1) \tag{13}$$

$$\sigma_i(k) = \alpha\sigma_i(k) + \beta\sigma_i(k-1) \tag{14}$$

where  $\alpha$  and  $\beta$  are a fixed weights of  $\mu_i(k)$  and  $\mu_i(k-1)$ . The learning method depending 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. In order to update the probability model in this paper, we adopt a different learning rule for  $\mu$  and  $\sigma$ .  $\mu$  is the candidate of optimum solution. Therefore, we hope it updated in real-time. Hence, we adopt a fast learning rule for  $\mu$  ( $\alpha=1, \beta=0$ ). The standard deviation  $\sigma$  should be a large value to search widely at beginning [34]. Thus, the  $\sigma$  is set a squared size of 1/2 of domain. In order to update the  $P(x)$ , the learning rule of  $\sigma$  is shown in equation (15).

$$\sigma_i(k) = w\sigma_i(k) + (1-w)\sigma_i(k-1) \tag{15}$$

$w$  is the learning rate. A bigger  $w$  can provide better real time ability of  $\sigma$ . A smaller  $w$  can ensure better robust of  $\sigma$ . In the paper, the value of  $w$  is adaptive to the iterations.

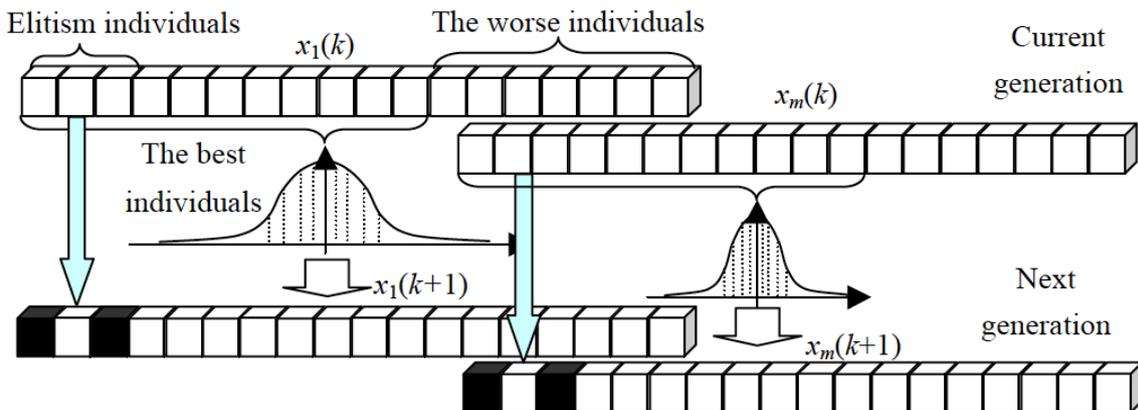


Fig. (3). Population operation diagram.

$$w = w_{\min} \left( \frac{w_{\max}}{w_{\min}} \right)^{\frac{FEs-1}{Max\_FEs-1}} \tag{16}$$

where  $w_{max}$  and  $w_{min}$  are the maximal and minimal value of  $w$ .  $FES$  is the current iterations, and  $Max\_FES$  is the maximal iterations.

### 3.2. Elitism Strategy

Elitism strategy is an effective strategy to ensure the best individual(s) is selected as the next generation in EAs, because the best individual(s) maybe include the information of optimal solution. Therefore, elitism can improves the convergence performance of EAs in many cases [38], and elitism has long been considered an effective method for improving the efficiency of EAs [39]. This is achieved by simply copying the best individual(s) directly to the new generation [40]. However, the number of best individuals selected as the next generation must be handled properly and carefully otherwise may lead to premature convergence or can not improve the efficiency of algorithm. Fig. (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 next generation.

$$Pop(k+1) = Elitism(k)_{NB} \cup Sample(k)_{NP-NB} \tag{17}$$

where  $Elitism()$  is the operator to copy the best solution to  $Pop(k+1)$ , and  $Sample()$  is the sampling function.  $N$  is the population size,  $NB$  is the number of best individuals selected to build probability model.

### 3.3. Local Search Strategy

It is widely accepted that a local search procedure is efficient in improving the solutions generated by the EDA. Kinds of strategies are proposed to enhance the performance of EDA, including the combing with PSO [4] and DE [18], quantum EDA [13], etc. In this paper, we use a chaos perturbation as the local search strategy. The principle of perturbation is shown as Fig. (4).

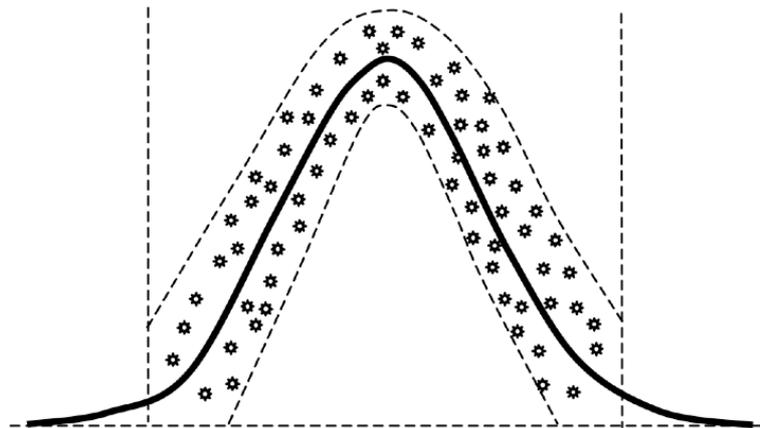


Fig. (4). The principle of perturbation.

The running of chaos operator is conditional. In this paper, the chaos perturbation is running under the condition of slower convergence.

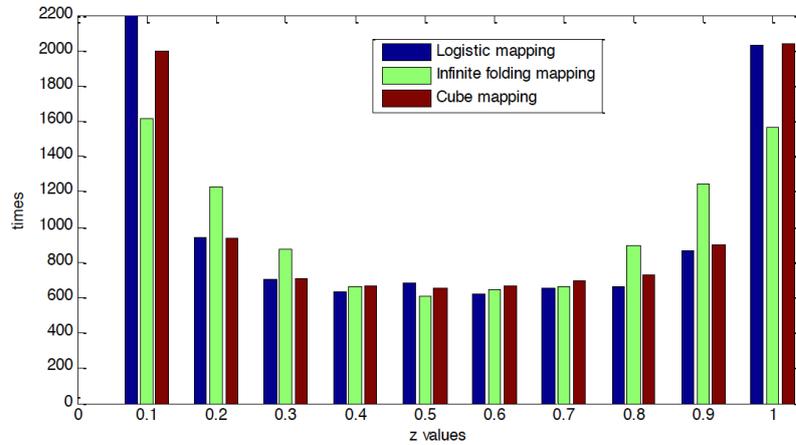
$$\begin{cases} Pop_i^j(k) = Pop_i^j(k) + \eta z_i & \text{if } < \text{meet criteria} > \\ Pop_i^j(k) = Pop_i^j(k) & \text{else } < \text{does not meet criteria} > \end{cases} \tag{18}$$

where  $Pop_i^j(k)$  is the  $i$ -th variable of  $j$ -th individual of  $k$ -th iteration.  $\eta$  is a perturbation coefficient.  $z_i$  is the chaotic variable, which can be generated by chaotic models. There are many chaotic models to generate the chaotic variables [41], such as Logistic mapping, Cube mapping or infinite folding mapping. Logistic chaotic model is the most popular one, which folded within a limited number under a limited range [42]. The logistic model is shown as follows.

$$z_{k+1} = \mu z_k (1 - z_k), \quad n = 0, 1, 2, 3... \tag{19}$$

It is a typical chaotic system. is the control variable, and a definite time series can be generated by iteration for any  $z_0 \in [0, 1]$ .

The model of infinite folding mapping is shown as equation (20):



**Fig. (5).** Distribution property of logistic, cube and infinite folding mapping.

$$z_{k+1} = \begin{cases} \sin \frac{2}{z_k} & k = 0, 1, 2 \dots \\ -1 < z_0 < 1 \end{cases} \tag{20}$$

The cube mapping is shown as equation (21)

$$z_{k+1} = \begin{cases} 4z_k^3 - 3z_k & k = 0, 1, 2 \dots \\ -1 < z_0 < 1 \end{cases} \tag{21}$$

The Cube and infinite folding mapping do not need control variable  $u$ . The distribution of chaotic sequence in logistic mapping is asymmetric, which follows the property of Chebyshev. The three mappings have 10000 iterations, and the statistics the results of the distribution properties of the mappings are shown in Fig. (5). The distribution property of infinite folding mapping is better than other one. Therefore, the infinite folding mapping is adopted to generate the chaotic variables. Additionally, the infinite folding mapping is insensitivity to the initial value [42].

The above way to generate new solutions does not take into account the feasibility of the solutions. The individuals could be out of the domain due to the perturbation. Therefore, a repair procedure is needed if illegal individuals are constructed.

$$\begin{cases} Pop_i^j(k) = Pop_i^j(k) & \text{if } Pop_i^j(k) \in [lb_i, ub_i] \\ Pop_i^j(k) = ub_i & \text{or } Pop_i^j(k) = lb_i, \text{ else } Pop_i^j(k) > ub_i \\ & \text{or } Pop_i^j(k) < lb_i \end{cases} \tag{22}$$

$[lb_i, ub_i]$  is the domain of  $i$ -th variable.

### 3.4. Procedure of ALREEDA

With the design above, the procedure of the ALREEDA is illustrated as following.

**Begin**

*Initialization:* Set parameters:  $Max\_FEs$ ,  $NP$ ,  $NB$ ,  $w\_max$ ,  $w\_min$ ,  $\sigma_i^0$ ,  $[lb_i\ ub_i]$ , and generate population  $Pop(0)$ .

**While** (stop criteria ?)

Generate  $w$  according FEs.

*Evaluation:* Calculate the fitness of all individuals, and store the elitism.

*Statistical information obtaining:* Select  $NB$  individuals to estimate the parameter of the probabilistic model, and update the parameter.

$$\mu_i(k) = \frac{1}{N} \sum_{n=1}^{NB} x_i^n(k)$$

$$\sigma_i^2(k) = \frac{1}{N} \sum_{n=1}^{NB} (x_i^n(k) - \mu_i(k))(x_i^n(k) - \mu_i(k))^T$$

$$\sigma_i(k) = w\sigma_i(k) + (1-w)\sigma_i(k-1)$$

*Probabilistic model building:* According to estimated parameters, build the probabilistic model of each variable  $x_i$

$$P(x_1, x_2, \dots, x_m) = \prod_{i=1}^m \frac{1}{\sigma_i^k \sqrt{2\pi}} e^{-\frac{(x_i - \mu_i^k)^2}{2\sigma_i^{k2}}}$$

*Probabilistic sampling:*

Make use of the sampling technology sampling ( $NP-NB$ ) individuals.

*Elitism strategy:* Combine the sampling individuals with elitism and generate new  $Pop(k)$ .

$$Pop(k) = Elitism(NB)_k \cup Sample(N - NB)_k$$

*Chaos perturbation* (perturbation criteria ?):

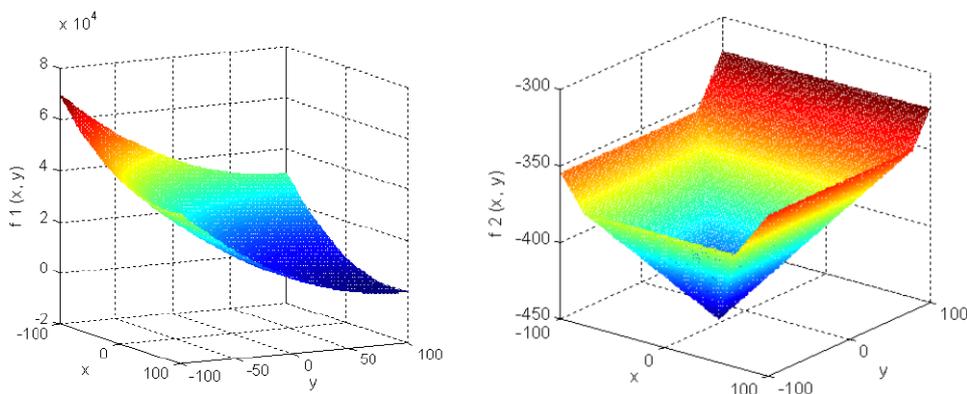
$$Pop_i^j(k) = Pop_i^j(k) + \eta z_i$$

$$k = k + 1$$

**End While**

**End Begin**

In summary, in the iteration of ALREEDA, the elitism strategy can maintain a better convergent performance, and the adaptive learning rate of  $\sigma$  will make the  $\sigma$  adaptive to the requirement of the algorithm. In the optimization, the promising region of the solution space may be found by the EDA. Then, the chaos perturbation search strategy can search the promising region to obtain better solutions. The benefits of the EDA and the local search are combined to balance global exploration and local exploitation.



**Fig. (6).** The graph of  $f1$  and  $f2$ .

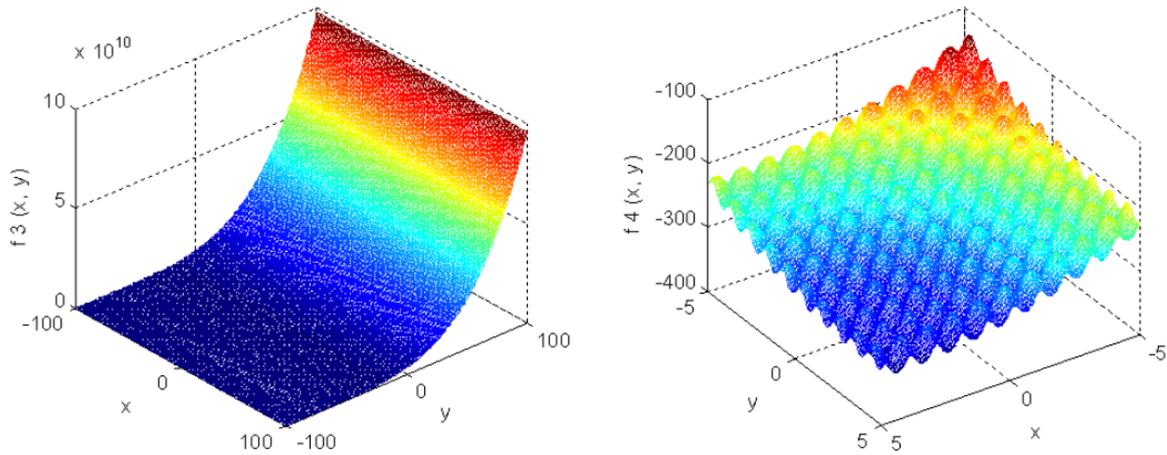


Fig. (7). The graph of  $f_3$  and  $f_4$ .

#### 4. SIMULATION EXPERIMENTS

To testify the performance and scalability of the proposed algorithm, the seven benchmarks of CEC'08 special session on large scale global optimization are adopted. The testing suites have different features, such as unimodal or multimodal, separable or non separable. In order to prevent search the symmetrical domain and the typical zero global optimum, the global optimum location of classical functions are shifted to a certain place different from zero and the global minimum are not zero.  $f_1$  is a shifted Sphere function,  $f_2$  is a shifted schwefel function,  $f_3$  is a shifted Rosenbrock function,  $f_4$  is a shifted Rastrigin function,  $f_5$  is a shifted Griewank function, and  $f_6$  is shifted Ackley function.  $f_7$  is a special function with complex structure. The concrete formulas of functions are shown as follows, and we also give a graphical expositions to express the complexity of the functions.

$$f_1(\mathbf{x}) = \sum_{i=1}^D (x_i - o_i)^2 + bias ,$$

$\mathbf{x} = (x_1, x_2, \dots, x_D)$ ,  $x_i \in [-100, 100]$ ,  $D$  is the dimensions.  $(o_1, o_2, \dots, o_D)$  is the shifted global optimum.  $f_1$  is an unimodal separable function. The global minimal is -450 at point  $(0_1, 0_2, \dots, 0_D)$ .

$$f_2(\mathbf{x}) = \max_i \{|(x_i - o_i)|, 1 \leq i \leq D\} + bias ,$$

$\mathbf{x} = (x_1, x_2, \dots, x_D)$ ,  $x_i \in [-100, 100]$ ,  $D$  is the dimensions.  $(o_1, o_2, \dots, o_D)$  is the shifted global optimum.  $f_2$  is a unimodal non-separable function. The global minimal is -450 at point  $(0_1, 0_2, \dots, 0_D)$  as shown in Fig. (6).

$f_1$  and  $f_2$  are unimodal functions.  $f_1$  is a slanted plane, and the minimum is at tip of the bottom.  $f_2$  is a squared funnel, and it is a non separable function which is harder for optimization than  $f_1$ .

$$f_3(\mathbf{x}) = \sum_{i=1}^{D-1} 100(((x_i - o_i + 1)^2 - (x_{i+1} - o_{i+1} + 1))^2 + (x_i - o_i)^2) + bias ,$$

$\mathbf{x} = (x_1, x_2, \dots, x_D)$ ,  $x_i \in [-100, 100]$ ,  $D$  is the dimensions.  $(o_1, o_2, \dots, o_D)$  is the shifted global optimum.  $f_3$  is a multimodal non separable function. The global minimal is 390 at point  $(0_1, 0_2, \dots, 0_D)$ .

$$f_4(\mathbf{x}) = \sum_{i=1}^D ((x_i - o_i)^2 - 10 \cos(2\pi(x_i - o_i)) + 10) + bias ,$$

$\mathbf{x} = (x_1, x_2, \dots, x_D)$ ,  $x_i \in [-5, 5]$ ,  $D$  is the dimensions.  $(0_1, 0_2, \dots, 0_D)$  is the shifted global optimum.  $f_4$  is a multimodal separable function with huge local minimal. The global minimal is -330 at point  $(0_1, 0_2, \dots, 0_D)$  as shown in Fig. (7).

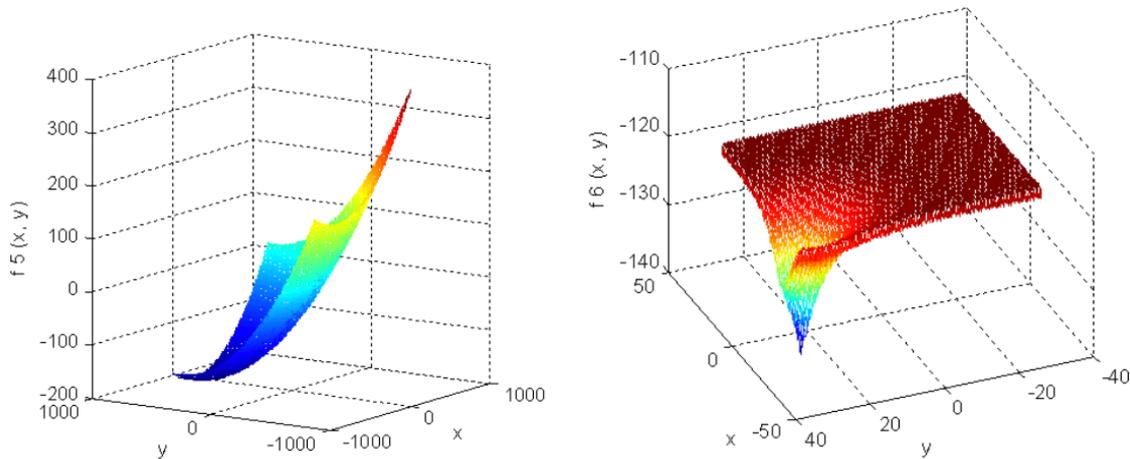


Fig. (8). The graph of  $f_5$  and  $f_6$ .

$f_3$  and  $f_4$  are multimodal functions as shown in.  $f_3$  is a slanted and curved surface. The minimum of  $f_3$  is located at wide and plane zone with similar value.  $f_4$  is slanted and waved surface, and there are huge minimum on the surface.  $f_3$  is a non-separable function and  $f_4$  is a complex function, which are a challenge for any algorithm.

$$f_5(\mathbf{x}) = \sum_{i=1}^D \frac{(x_i - o_i)^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i - o_i}{\sqrt{i}}\right) + 1 + bias,$$

$\mathbf{x} = (x_1, x_2, \dots, x_D)$ ,  $x_i \in [-600, 600]$ ,  $D$  is the dimensions.  $(0_1, 0_2, \dots, 0_D)$  is the shifted global optimum.  $f_5$  is a multimodal non-separable function. The global minimal is -180 at point  $(0_1, 0_2, \dots, 0_D)$ .

$$f_6(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D (x_i - o_i)^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi(x_i - o_i))\right) + 20 + e + bias,$$

$\mathbf{x} = (x_1, x_2, \dots, x_D)$ ,  $x_i \in [-32, 32]$ ,  $D$  is the dimensions.  $(0_1, 0_2, \dots, 0_D)$  is the shifted global optimum.  $f_6$  is a multimodal separable function. The global minimal is -140 at point as shown in Fig. (8).

$f_5$  is a slanted plane, and the minimum is located at the bent bottom.  $f_6$  has graphic of conical funnel.  $f_5$  and  $f_6$  are easier for optimization.

$$f_7(\mathbf{x}) = \sum_{i=1}^D fractal1D(x_i + twist(x_{(i \bmod D)+1}))$$

$$twist(y) = 4(y^4 - 2y^3 + y^2)$$

$$Fractal1D(x) \approx \sum_{k=1}^3 \sum_{l=1}^{2^{k-1}} \sum_{m=1}^{ran2(o)}$$

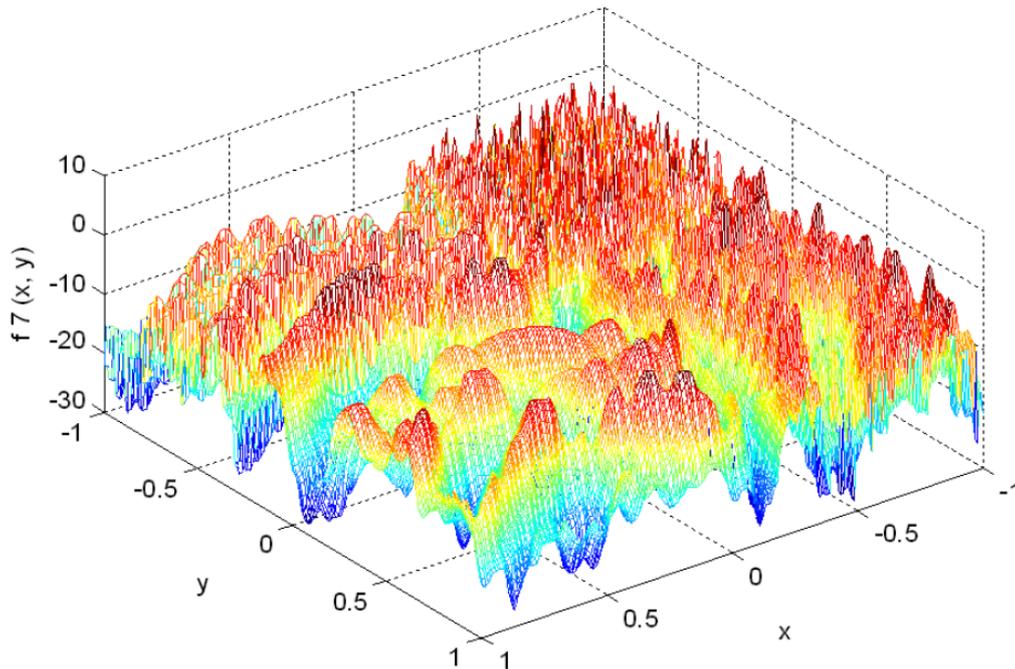
$$doubledip(x, ran1(o), \frac{1}{2^{k-1}(2 - ran1(o))})$$

$$doubledip(x, c, s) =$$

$$\begin{cases} (-6144(x-c)^6 + 3088(x-c)^4 - 392(x-c)^2 + 1)s, & x \in (-0.5, 0.5) \\ 0, & \text{otherwise} \end{cases}$$

$\mathbf{x} = (x_1, x_2, \dots, x_D)$ ,  $D$  is the dimensions,  $x_i \in [-1, 1]$ .  $ran1(o)$  is a pseudorandomly chosen function, with seed  $o$  and equal probability from the interval  $[10, 1]$ , and having the precision of double.  $ran2(o)$  is also pseudorandomly chosen function, with seed  $o$  and equal probability from the set  $\{0, 1, 2\}$ .  $fractal1D(x)$  is an approximation to a recursive

algorithm, it does not take account of wrapping at the boundaries, or local re-seeding of the random generators.  $f7$  is a multimodal non-separable function. The global minimal is unknown as shown in Fig. (9).



**Fig. (9).** The graph of  $f7$ .

We can see from the graph of  $f7$ ,  $f7$  is a very complex function. It looks like mountain range profile. There are many bigger maintains and also many narrow and dense peaks. The global optimum of  $f7$  is unknown so far. The function  $f7$  always takes negative value. In this case, we use  $(f_7(x) - f_7(x^*))$  as the result value. Therefore, the  $f7$  values of figures will be negative.

#### 4.1. The Selection of NP and NB

In the iteration optimization algorithm, such as EAs, the maximal iteration number (Max\_FEs) is an essential parameter [19], while it is maybe varied. The Max\_FEs is set to  $3E+6$  in this paper. For EDAs, the population size NP and the promising solution number NB are important except for Max\_FEs. It is obvious that for an easy problem, a small value of NP is sufficient, but for difficult problems, a large value of NP is recommended in order to avoid trapping to a local optimum. The large NP may provide better optimization with larger calculation [16]. However, it may vary from problem to problem. We pay attention to the performance of EDA instead of the population size. We provide some choices (100, 200, 300, 500, 1000) to obtain better performance for the ALREEDA. We have a comparisons of different population size, and select the best population size as the final decision of the algorithm on the problem with the given problem size. The NB is also an important parameter for the probabilistic model learning of EDA, and we also have a comparison to determine a proper NB. In the comparisons, the error recorded finally is the absolute margin between the fitness of the best solution found and the fitness of the global optimum. The semi-log graphs show  $\log_{10}(\text{error})$  vs. FES for the first 6 functions.

In Fig. (10), it is the testing result when the dimension D is 100 and has different NP and NB. We can see from the figure, the optimization results are similar for  $f_1$ ,  $f_3$ ,  $f_5$ ,  $f_6$  and  $f_7$  when the population size is 100, 200, 300 or 500. According to  $f_2$  and  $f_4$ , it is a proper choice that the population size is 500. Further more, the algorithm have a better performance when NP is 500 and NB is 50.

We also do some tests when the dimension D is 500. Firstly, we test the algorithm under different population size 100, 200 or 500. From Fig. (11), we can see the performance of the algorithm on  $f_1$ ,  $f_3$ ,  $f_5$  and  $f_6$  are similar regardless of population size. It only effect the optimization time. According to the performance of the algorithm on  $f_2$ ,  $f_4$  and  $f_7$ , the population size is selected as 500. 50 is a suitable value for NB according to the comprehensive performance of the algorithm on the benchmarks.

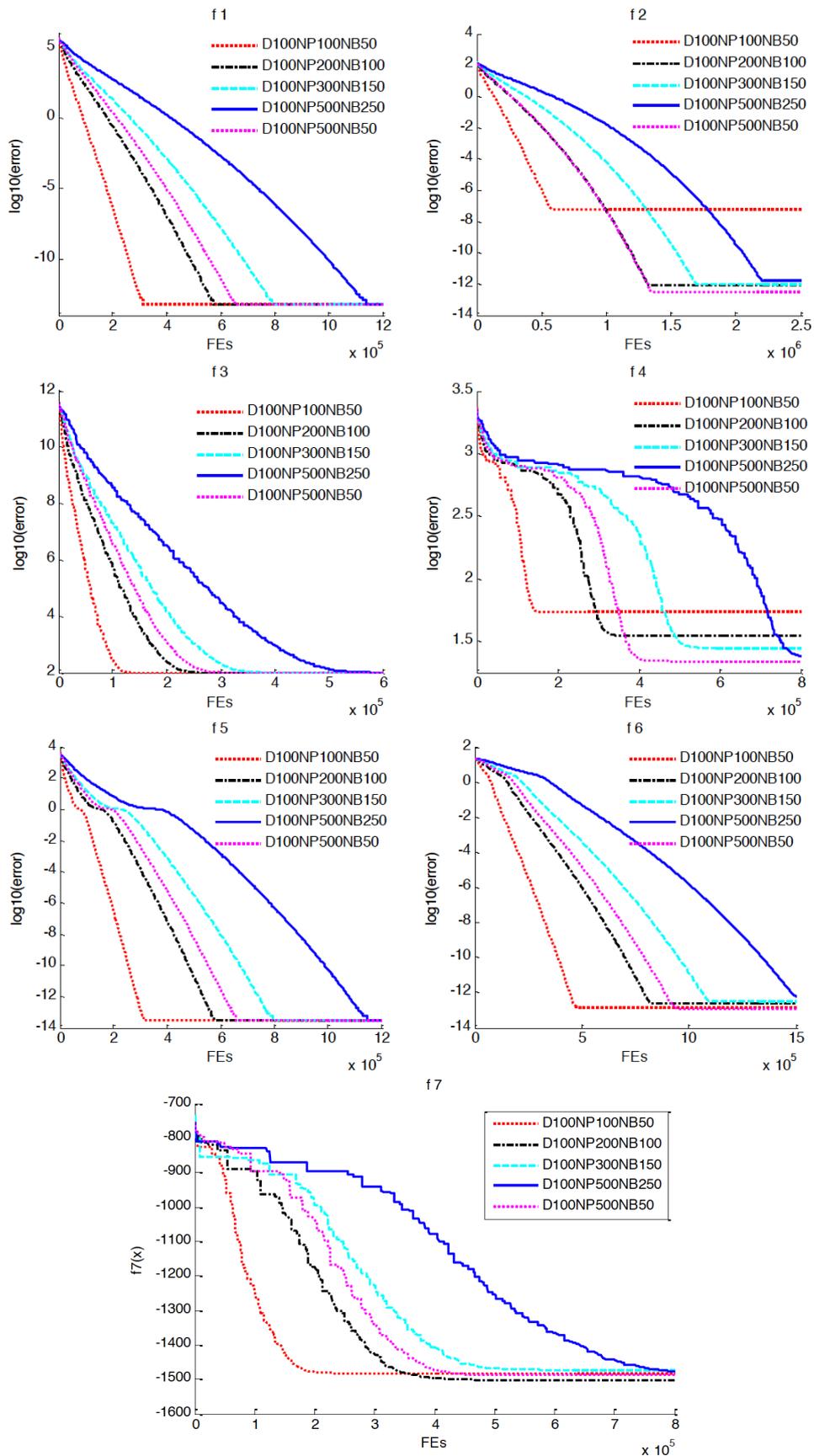


Fig. (10). Convergence graph for function 1-7 (D=100, different NP).

Due to the time reason, we didn't do the population size test for  $D=1000$ . We testify the algorithm directly on  $NP=500$  and  $NB=50$ . Fig. (12) shows the performance of the algorithm, and also have a comparison of different dimension size when  $NP=500$  and  $NB=50$ .

According to Fig. (12), we also can conclude that  $f1$ ,  $f5$  and  $f6$  are easy to solve for different dimension sizes. However, the performance is promising for  $f2$  when  $D=100$ , the results are worse when  $D$  is 500 or 1000 with  $D=100$ . For  $f3$  and  $f4$ , the error is increasing with the dimension.

From the above mentioned, the population size is selected as 500, and  $NB$  is 50 for dimension 100, 500 and 1000. We also can conclude that the ALREEDA is scale to the dimension size except for the  $f2$ .

We also do a small test to testify the effect of the elitism strategy. The optimization process partial enlarge graph of  $f7$ ,  $f4$ , and  $f2$  are shown in Fig. (13).  $f7$  is the most obvious one. The optimization process is concussive without elitism strategy due to the complex  $f7$  function, though the optimization is convergent finally. The stability of optimization process for  $f4$  is better, and it only has a small shock. For  $f2$ , it has tiny fluctuation due to the characteristics of function. The convergence is smooth and steady when the elitism strategy is added to the algorithm.

## 4.2. Comparison with EDAs

In this section, we compare ALREEDA with LSEDA-gl. LSEDA-gl is a robust univariate EDA, which is proposed for large scale optimization and has good performance. In LSEDA-gl, an effective sampling under mixed Gaussian and Levy probability distribution is introduced to balance optimization and learning. And a restart mechanism is used to stop some variables shrinking dramatically to zero solely. The optimization results of 100-D, 500-D, and 1000-D are indicated in Table 1. The results of ALREEDA on  $f3$ ,  $f4$ , and  $f7$  outperform LSEDA-gl. The results of the two algorithms are similar on  $f1$ ,  $f5$  and  $f6$ . For  $f2$ , the two algorithms are similar on 100-D. However, the performance of LSEDA-gl is better than ALREEDA on 500-D and 1000-D. The better and comparable results are marked by bold type in Table 1.

**Table 1. Error values for functions 1-7 with  $D=100$ , 500 and 1000.**

Algorithms	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$
D=100							
LSEDA-gl	<b>5.68E-14</b>	<b>2.21E-13</b>	2.81E+02	1.31E+02	<b>2.84E-14</b>	<b>9.78E-14</b>	-1.46E+03
ALREEDA	<b>5.68E-14</b>	1.53E-12	<b>9.61E+01</b>	<b>1.99E+01</b>	<b>0.00E-14</b>	<b>5.96E-13</b>	<b>-1.50E+03</b>
D=500							
LSEDA-gl	<b>2.27E-13</b>	<b>2.72E-10</b>	<b>8.67E+02</b>	<b>8.55E+02</b>	<b>1.14E-13</b>	<b>3.13E-13</b>	-6.83E+03
ALREEDA	<b>1.14E-13</b>	7.27E+00	<b>4.92E+02</b>	<b>2.32E+02</b>	<b>5.68E-13</b>	<b>1.99E-13</b>	<b>-7.15E+03</b>
D=1000							
LSEDA-gl	<b>3.22E-13</b>	<b>1.04E-05</b>	1.73E+03	<b>5.45E+02</b>	<b>1.71E-13</b>	<b>4.26E-13</b>	-1.35E+04
ALREEDA	<b>2.27E-13</b>	3.93E+01	<b>8.01E+02</b>	<b>6.53E+02</b>	<b>1.14E-13</b>	<b>3.41E-13</b>	<b>-1.40E+04</b>

The convergent graph on 1000-D of the two algorithms is shown in Fig. (14). From the figure we can see the convergent process of ALREEDA is gentle than LSEDA-gl. In comparison, ALREEDA outperforms LSEDA-gl on  $f3$ ,  $f4$  and  $f7$  even though the convergence is kindly. However, the performance of ALREEDA is wore on  $f2$  than LSEDA-gl on 1000-D.

## 4.3. Comparison with other Algorithms

In order to indicate the performance of the algorithm, we compare the algorithm with other state of the art algorithms, like the cooperative coevolution DE algorithm MLCC [10], the dynamic multi-swarm particle swarm optimizer DMS-PSO [43], the efficiency population utilization strategy based PSO EPUS-PSO [44], the improved PSO DEWSAcc [45] and jDEdynNP-F [46], and the multiple trajectory search algorithm MTS [47], which took part in the competition of CEC'08 on large scale optimization. According to the competition result, a comparison is shown in Table 2 on  $D=100$ , 500 and 1000. In the table, in order to express the results clearly, the better and comparable results are marked by bold type.

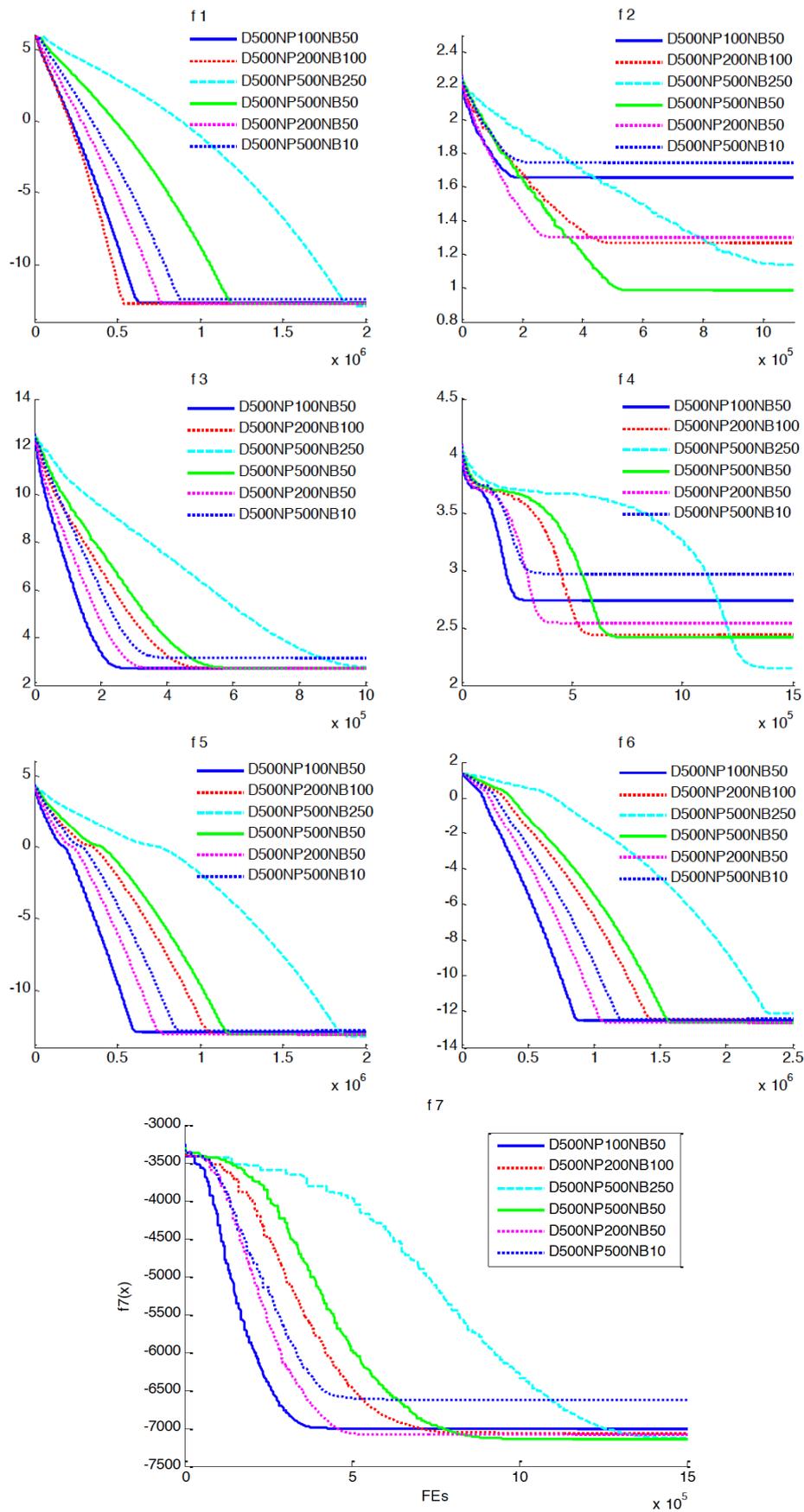
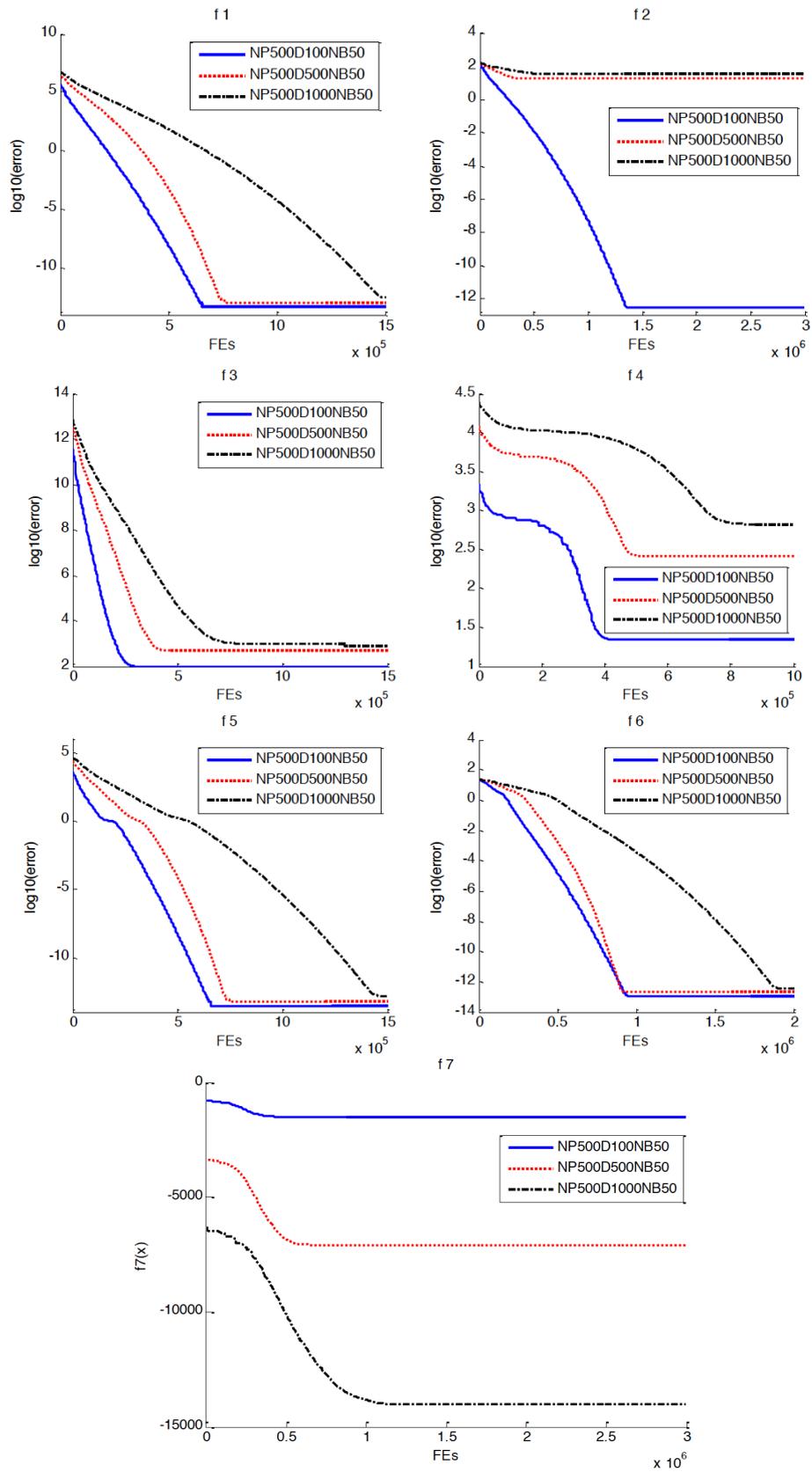


Fig. (11). Convergence graph for function 1-7 (D=500, different NP).



**Fig. (12).** Convergence graph for function 1-7 (different D, the same NB).

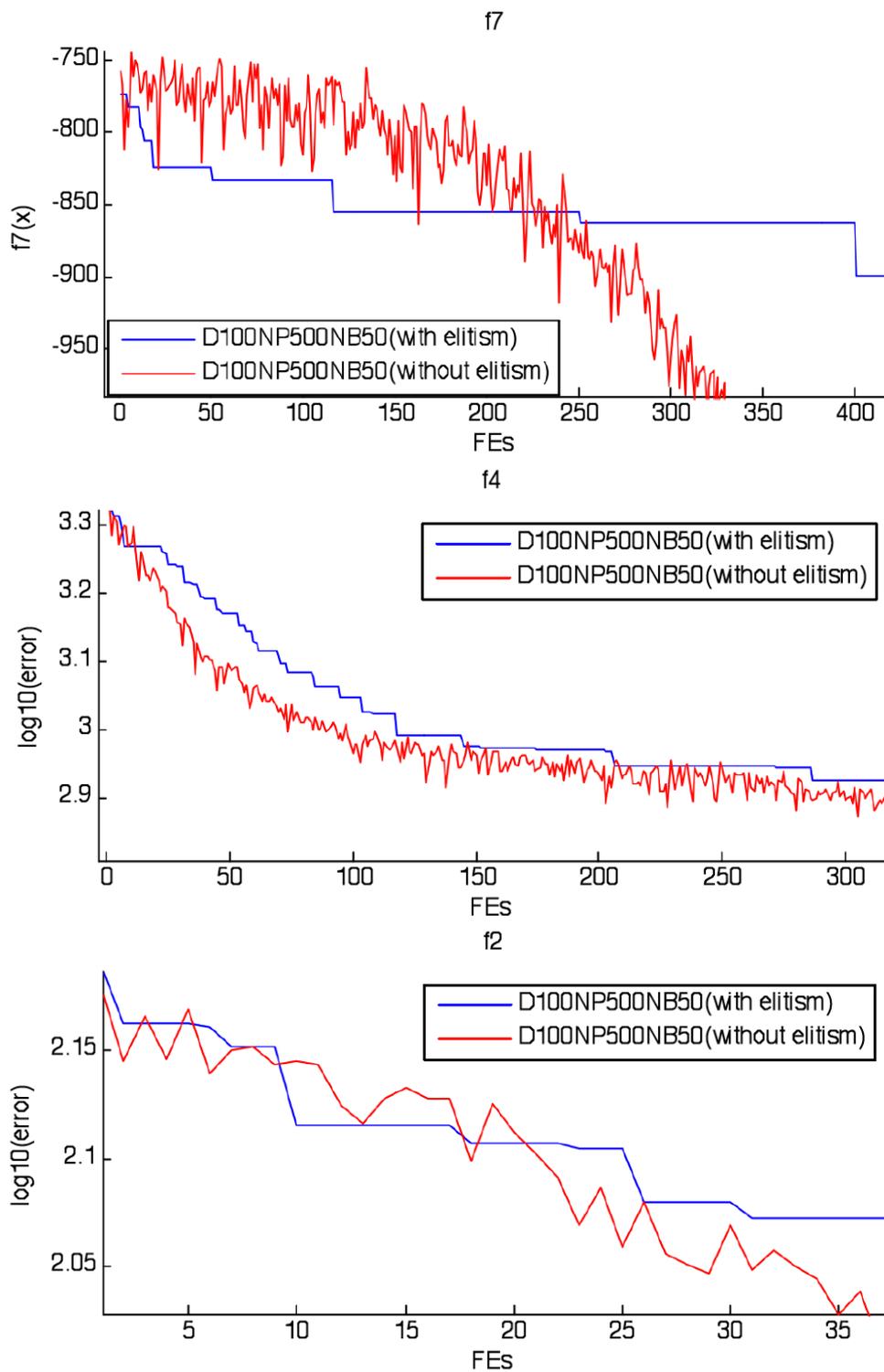
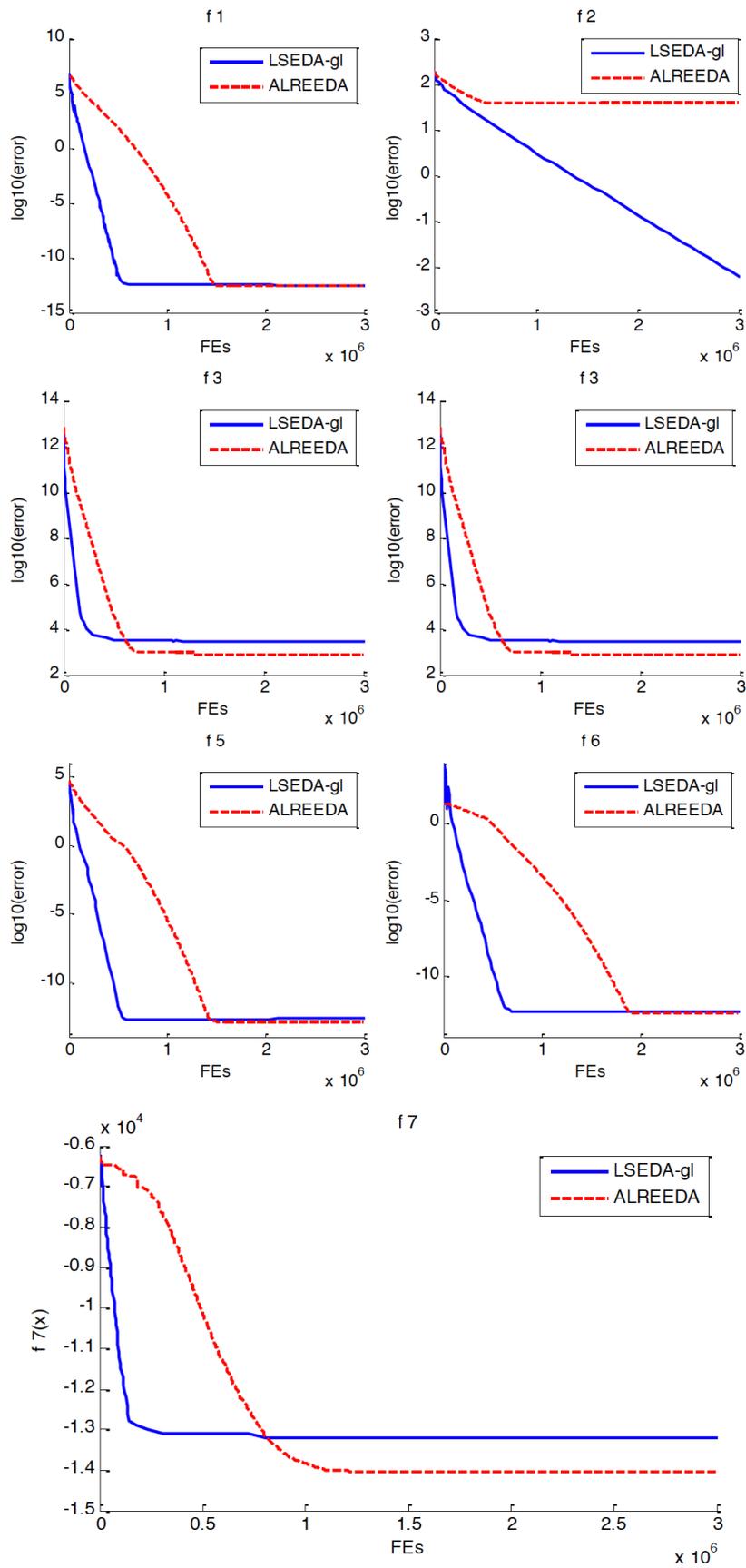


Fig. (13). The optimization process partial enlarge graph of  $f_7$ ,  $f_4$  and  $f_2$ .



**Fig. (14).** Convergence graph for function 1-7 (D=1000).

Table 2. Error values for functions 1-7 with D=100, 500 and 1000.

Algorithms	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$
D=100							
MLCC	<b>6.82E-14</b>	2.53E+01	1.49E+02	<b>4.39E-13</b>	<b>3.41E-14</b>	<b>1.11E-13</b>	<b>-1.54E+03</b>
EPUS-PSO	7.47E-01	1.86E+01	4.99E+03	4.71E+02	3.72E-01	2.06E+00	-8.55E+02
JDEdynNP-F	<b>5.68E-14</b>	4.29E-01	1.12E+02	<b>5.46E-14</b>	<b>2.84E-14</b>	<b>5.68E-14</b>	-1.48E+03
MTS	0.00E+00	<b>1.44E-11</b>	<b>5.17E-08</b>	0.00E+00	0.00E+00	0.00E+00	-1.49E+03
DewSAcc	<b>5.68E-14</b>	8.25E+00	1.45E+02	4.38E+00	<b>3.07E-14</b>	<b>1.13E-13</b>	-1.37E+03
DMS-PSO	0.00E+00	3.64E+00	2.83E+01	1.83E+02	0.00E+00	0.00E+00	-1.14E+03
ALREEDA	<b>5.68E-14</b>	<b>1.53E-12</b>	9.61E+01	1.99E+01	<b>0.00E-14</b>	<b>5.96E-13</b>	<b>-1.50E+03</b>
D=500							
MLCC	<b>4.30E-013</b>	6.67E+01	9.25E+02	<b>1.79E-11</b>	<b>2.13E-13</b>	<b>5.34E-13</b>	-4.44E+03
EPUS-PSO	8.45E+01	4.35E+01	5.77E+04	3.49E+03	1.64E+00	6.64E+00	-3.51E+03
JDEdynNP-F	<b>9.32E-14</b>	8.46E+00	6.61E+02	<b>1.47E-12</b>	<b>4.21E-14</b>	<b>1.49E-13</b>	-6.88E+03
MTS	0.00E+00	<b>7.32E-06</b>	<b>5.03E-03</b>	0.00E+00	0.00E+00	<b>6.18E-12</b>	<b>-7.08E+03</b>
DewSAcc	2.09E-09	7.57E+01	1.81E+03	3.64E+02	6.90E-04	4.80E-01	-5.75E+03
DMS-PSO	0.00E+00	6.89E+01	4.67E+07	1.61E+03	0.00E+00	2.00E+00	-4.20E+03
ALREEDA	<b>1.14E-13</b>	7.27E+00	4.92E+02	2.32E+02	<b>5.68E-13</b>	<b>1.99E-13</b>	<b>-7.15E+03</b>
D=1000							
MLCC	<b>8.46E-13</b>	1.09E+02	1.80E+03	<b>1.37E-10</b>	4.18E-03	<b>1.06E-12</b>	<b>-1.47E+04</b>
EPUS-PSO	5.53E+02	4.66E+01	8.37E+05	7.58E+03	5.89E+00	1.89E+01	-6.62E+03
JDEdynNP-F	<b>1.14E-13</b>	1.95E+01	1.31E+03	2.17E-04	<b>3.98E-14</b>	<b>1.47E-11</b>	-1.35E+04
MTS	0.00E+00	<b>4.72E-02</b>	<b>3.41E-04</b>	0.00E+00	0.00E+00	<b>1.24E-11</b>	<b>-1.40E+04</b>
DewSAcc	8.79E-03	9.61E+01	9.15E+03	1.82E+03	3.58E-03	2.30E+00	-1.06E+04
DMS-PSO	0.00E+00	9.15E+01	8.98E+09	3.84E+03	0.00E+00	7.76E+00	-7.51E+03
ALPSEA	3.58E+04	1.47E+02	2.33E+09	1.89E+02	3.04E+02	1.13E+01	N/A
ALPSEA-100M	1.80E-05	9.30E+01	1.77E+03	1.53E-02	1.17E-06	4.40E-04	N/A
ALREEDA	<b>2.27E-13</b>	3.93E+01	8.01E+02	6.53E+02	<b>1.14E-13</b>	<b>3.41E-13</b>	<b>-1.40E+04</b>

ALREEDA has a good performance on  $f_1, f_5, f_6$  and  $f_7$ , and it is comparable with the MLCC and JDEdynNP-F with dimensions 100, 500 and 1000. For function  $f_2$ , ALREEDA has a good performance than other algorithms on D=100. However, it becomes worse suddenly, although the results are comparable with other algorithms on D=500 and 1000. This may be caused by the using of simple probabilistic model, and also  $f_2$  is a non-separable function. MTS has a better performance than other algorithms for  $f_2, f_3$  and  $f_4$  are multimodal functions with separable and non-separable characteristic, which are a challenge for any algorithms. For function  $f_3$ , MTS has a good performance. MLCC outperformed other algorithms with high precisions on  $f_4$ , especially for D=1000. Although the ALREEDA is not the best solver for  $f_3$  and  $f_4$ , it has the similar performance as others, and it scales well on D=100, 500 and 1000.  $f_7$  is a very complex function, the performance of ALREEDA is promising.

## CONCLUSION

In this paper, we present the adaptive learning rate elitism EDA combining a chaos perturbation search strategy. We demonstrate the ALREEDA's ability in scaling up to large scale optimization problem. Several strategies have been designed to enhance the large scale optimization capability of ALREEDA. The performance of ALREEDA is promising, and it is scale to dimensions (100, 500 and 1000). Furthermore, ALREEDA was compared with the LSEDA-gl, and other algorithms. ALREEDA is competitive on  $f_1, f_5, f_6$  and  $f_7$ , though the LSEDA-gl has a good performance on  $f_2$ . Compared with other algorithms proposed on CEC'08, the performance of ALREEDA on  $f_1, f_5, f_6$  and  $f_7$  is promising, though other algorithms (MLCC, JDEdynNP-F, MTS) perform better on  $f_2, f_3$  and  $f_4$ . According to the test on dimension 100, 500 and 1000, the ALREEDA is scale to the dimension size. It is just the beginning of us on large scale optimization research. In future, we are planning to examine more effective strategies to enhance the large scale optimization ability of EDAs.

## CONFLICT OF INTEREST

The authors of the manuscript do not have a direct financial relation with the commercial identity mentioned in this paper that might lead to a conflict of interest for any of the authors.

## ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61174044, Natural Science Foundation of Shandong Province under Grant No. ZR2015PF009 and Independent Innovation Foundation of Shandong University under grant No. 2015ZQXM002. The authors want to thank Ke Tang for providing the source code of the benchmarks and competition result of CEC'08 large scale optimization at website (<http://nical.ustc.edu.cn/cec08ss.php>).

## REFERENCES

- [1] S. Kravanja, A. Soršak, and Z. Kravanja, "Efficient multilevel minlp strategies for solving large combinatorial problems in engineering", *Optim. Eng.*, vol. 4, no. 1-2, pp. 97-151, 2003.  
[<http://dx.doi.org/10.1023/A:1021812414215>]
- [2] D. Fayek, G. Kesidis, and A. Vannelli, "Non-linear game models for large-scale network bandwidth management", *Optim. Eng.*, vol. 7, no. 4, pp. 421-444, 2006.  
[<http://dx.doi.org/10.1007/s11081-006-0348-y>]
- [3] H. Tokos, and Z. Pintarič, "Development of a MINLP model for the optimization of a large industrial water system", *Optim. Eng.*, vol. 13, no. 4, pp. 625-662, 2012.  
[<http://dx.doi.org/10.1007/s11081-011-9162-2>]
- [4] C.W. Ahn, J. An, and J. Yoo, "Estimation of particle swarm distribution algorithms: Combining the benefits of PSO and EDAs", *Inf. Sci.*, vol. 192, no. 1, pp. 109-119, 2012.  
[<http://dx.doi.org/10.1016/j.ins.2010.07.014>]
- [5] B.A. Al-Sarray, and R.A. Al-Dabbagh, "Variants of hybrid genetic algorithms for optimizing likelihood ARMA model function and many of problems", In: *Evolutionary Algorithms*. Baghdad University: Iraq, 2011, pp. 219-246.  
[<http://dx.doi.org/10.5772/16141>]
- [6] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution", *Inf. Sci.*, vol. 178, no. 15, pp. 2985-2999, 2008.  
[<http://dx.doi.org/10.1016/j.ins.2008.02.017>]
- [7] B. Yu, Z. Yang, and B. Yao, "An improved ant colony optimization for vehicle routing problem", *Eur. J. Oper. Res.*, vol. 196, no. 1, pp. 171-176, 2009.  
[<http://dx.doi.org/10.1016/j.ejor.2008.02.028>]
- [8] B. Yu, and Z.Z. Yang, "An ant colony optimization model: The period vehicle routing problem with time windows", *Transp. Res., Part E Logist. Trans. Rev.*, vol. 47, no. 2, pp. 166-181, 2011.  
[<http://dx.doi.org/10.1016/j.tre.2010.09.010>]
- [9] M.N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization", *IEEE Trans. Evol. Comput.*, vol. 3, no. X, pp. XX-XX, 2013.
- [10] Z. Yang, K. Tang, and X. Yao, "Multilevel cooperative coevolution for large scale optimization", In: *IEEE Congress on Evolutionary Computation, CEC 2008*. IEEE Press: Hong Kong, China, 2008, pp. 1663-1670. Available from: <http://sci2s.ugr.es/sites/default/files/files/TematicWebSites/EAMHCO/contributionsCEC08/yang08mcc.pdf>
- [11] H. Mühlenbein, and G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters", In: *Proceedings of the 4<sup>th</sup> International Conference on Parallel Problem Solving from Nature*. London: UK, 1996, pp. 178-187.  
[[http://dx.doi.org/10.1007/3-540-61723-X\\_982](http://dx.doi.org/10.1007/3-540-61723-X_982)]
- [12] S. Shahraki, and M.R. Tutunchy, "Continuous gaussian estimation of distribution algorithm", In: *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*. Springer: Berlin, Heidelberg, 2013, pp. 211-218.  
[[http://dx.doi.org/10.1007/978-3-642-33042-1\\_23](http://dx.doi.org/10.1007/978-3-642-33042-1_23)]
- [13] M.E. Platel, S. Schliebs, and N. Kasabov, "Quantum-inspired evolutionary algorithm: a multimodel EDA", *IEEE Trans. Evol. Comput.*, vol. 13, no. 6, pp. 1218-1232, 2009.  
[<http://dx.doi.org/10.1109/TEVC.2008.2003010>]
- [14] J. Yang, H. Xu, and P. Jia, "Effective search for Pittsburgh learning classifier systems via estimation of distribution algorithms", *Inf. Sci.*, vol. 198, no. 0, pp. 100-117, 2012.  
[<http://dx.doi.org/10.1016/j.ins.2012.02.059>]
- [15] X. Huang, P. Jia, and B. Liu, "Controlling chaos by an improved estimation of distribution algorithm", *Math. Comput. Appl.*, vol. 15, no. 5, pp. 866-871, 2010.
- [16] H. Karshenas, R. Santana, C. Bielza, P. Larra, and N. Aga, "Regularized continuous estimation of distribution algorithms", *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2412-2432, 2013.  
[<http://dx.doi.org/10.1016/j.asoc.2012.11.049>]
- [17] S.I. Valdez, A. Hernández, and S. Botello, "A Boltzmann based estimation of distribution algorithm", *Inf. Sci.*, vol. 236, no. 0, pp. 126-137, 2013.

- [http://dx.doi.org/10.1016/j.ins.2013.02.040]
- [18] Y. Wang, B. Li, and T. Weise, "Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems", *Inf. Sci.*, vol. 180, no. 12, pp. 2405-2420, 2010.  
[http://dx.doi.org/10.1016/j.ins.2010.02.015]
- [19] W. Dong, T. Chen, P. Tino, and X. Yao, "Scaling up estimation of distribution algorithms for continuous optimization", *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 792-822, 2013.  
[http://dx.doi.org/10.1109/TEVC.2013.2247404]
- [20] L. Lozada-Chang, and R. Santana, "Univariate marginal distribution algorithm dynamics for a class of parametric functions with unitation constraints", *Inf. Sci.*, vol. 181, no. 11, pp. 2340-2355, 2011.  
[http://dx.doi.org/10.1016/j.ins.2011.01.024]
- [21] L. Marti, J. Garcia, A. Berlanga, and J.M. Molina, "Multi-objective optimization with an adaptive resonance theory-based estimation of distribution algorithm", *Ann. Math. Artif. Intell.*, vol. 68, no. 4, pp. 247-273, 2013.  
[http://dx.doi.org/10.1007/s10472-012-9303-0]
- [22] V.A. Shim, K.C. Tan, J.Y. Chia, and A. Al Mamun, "Multi-objective optimization with estimation of distribution algorithm in a noisy environment", *Evo. Comput.*, vol. 21, no. 1, pp. 149-177, 2013.  
[http://dx.doi.org/10.1162/EVCO\_a\_00066]
- [23] V.A. Shim, K.C. Tan, and C.Y. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem", *IEEE Trans. Syst. Man Cybern. C*, vol. 42, no. 5, pp. 682-691, 2012.  
[http://dx.doi.org/10.1109/TSMCC.2012.2188285]
- [24] L. Wang, S. Wang, and Y. Xu, "An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem", *Expert Syst. Appl.*, vol. 39, no. 5, pp. 5593-5599, 2012.  
[http://dx.doi.org/10.1016/j.eswa.2011.11.058]
- [25] Y. Wang, J. Xiang, and Z. Cai, "A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator", *Appl. Soft Comput.*, vol. 12, no. 11, pp. 3526-3538, 2012.  
[http://dx.doi.org/10.1016/j.asoc.2012.06.008]
- [26] U. Aickelin, E.K. Burke, and J. Li, "An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering", *J. Oper. Res. Soc.*, vol. 58, no. 12, pp. 1574-1585, 2006.  
[http://dx.doi.org/10.1057/palgrave.jors.2602308]
- [27] Y. Wang, and B Li, "A restart univariate estimation of distribution algorithm: sampling under mixed Gaussian and  $L^{\nu}$  probability distribution", In: *Proceedings on Evolutionary Computation*. IEEE: Hong Kong, China, 2008, pp. 3917-3924.  
[http://dx.doi.org/10.1109/CEC.2008.4631330]
- [28] Y. Wang, B. Li, and T. Weise, "Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems", *Inf. Sci.*, vol. 180, no. 12, pp. 2405-2420, 2010.  
[http://dx.doi.org/10.1016/j.ins.2010.02.015]
- [29] K. Ata, J. Bootkrajang, and R.J Durrant, "Towards large scale continuous EDA: a random matrix theory perspective", In: *Proceeding of the 15<sup>th</sup> Annual Conference on Genetic and Evolutionary Computation Conference*. Amsterdam: Netherlands, 2013, pp. 383-390.  
[http://dx.doi.org/10.1145/2463372.2463423]
- [30] W. Dong, and X. Yao, "Unified eigen analysis on multivariate Gaussian based estimation of distribution algorithms", *Inf. Sci.*, vol. 178, no. 15, pp. 3000-3023, 2008.  
[http://dx.doi.org/10.1016/j.ins.2008.01.021]
- [31] P.A. Bosman, and D Thierens, "Numerical optimization with real-valued estimation of distribution algorithms", In: *Scalable Optimization via Probabilistic Modeling*. Springer: Berlin, Heidelberg, 2006, pp. 91-120.  
[http://dx.doi.org/10.1007/978-3-540-34954-9\_5]
- [32] S. Muelas, A. Mendiburu, A. LaTorre, and J. Peña, "Distributed estimation of distribution algorithms for continuous optimization: How does the exchanged information influence their behavior?", *Inf. Sci.*, vol. 268, pp. 231-254, 2014.  
[http://dx.doi.org/10.1016/j.ins.2013.10.026]
- [33] J. Sun, Q. Zhang, and E.P. Tsang, "DE/EDA: A new evolutionary algorithm for global optimization", *Inf. Sci.*, vol. 169, no. 3-4, pp. 249-262, 2005.  
[http://dx.doi.org/10.1016/j.ins.2004.06.009]
- [34] 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", *Proc. Comput. Sci.*, vol. 4, pp. 1244-1251, 2011.  
[http://dx.doi.org/10.1016/j.procs.2011.04.134]
- [35] L. Marti, J. Garcia, A. Berlanga, C.A. Coello, and J.M. Molina, "MB-GNG: Addressing drawbacks in multi-objective optimization estimation of distribution algorithms", *Oper. Res. Lett.*, vol. 39, no. 2, pp. 150-154, 2011.  
[http://dx.doi.org/10.1016/j.orl.2011.01.002]
- [36] P.A. Bosman, and J. Grahl, "Matching inductive search bias and problem structure in continuous estimation of distribution algorithms", *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1246-1264, 2008.  
[http://dx.doi.org/10.1016/j.ejor.2006.06.051]

- [37] 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*. Orlando: USA, 1999, pp. 840-846.
- [38] C.W. Ahn, and R.S. Ramakrishna, "Elitism-based compact genetic algorithms", *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 367-385, 2003. [<http://dx.doi.org/10.1109/TEVC.2003.814633>]
- [39] 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*. New York: USA, 2002, pp. 520-527.
- [40] I.J. Leno, S.S. Sankar, M.V. Raj, and S.G. Ponnambalam, "An elitist strategy genetic algorithm for integrated layout design", *Int. J. Adv. Manuf. Technol.*, vol. 66, no. 9-12, pp. 1573-1589, 2013.
- [41] M.S. Tavazoei, and M. Haeri, "Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms", *Appl. Math. Comput.*, vol. 187, no. 2, pp. 1076-1085, 2007. [<http://dx.doi.org/10.1016/j.amc.2006.09.087>]
- [42] Q. Xu, S. Wang, L. Zhang, and Y. Liang, "A novel chaos danger model immune algorithm", *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 11, pp. 3046-3060, 2013. [<http://dx.doi.org/10.1016/j.cnsns.2013.04.017>]
- [43] S. Zhao, J.J. Liang, P.N. Suganthan, and M.F. Tasgetiren, "Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization", In: *Proceedings on Evolutionary Computation*. IEEE: Hong Kong, China, 2008, pp. 3845-3852. [<http://dx.doi.org/10.1109/CEC.2008.4631320>]
- [44] S. Hsieh, T. Sun, C. Liu, and S. Tsai, "Solving large scale global optimization using improved particle swarm optimizer", In: *Evolutionary Computation*. IEEE: Hong Kong, China, 2008, pp. 1777-1784.
- [45] A. Zamuda, J. Brest, B. Boskovic, and V. Zumer, "Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution", In: *IEEE Congress on Evolutionary Computation*. IEEE: Hong Kong, 2008, pp. 3718-3725. [<http://dx.doi.org/10.1109/CEC.2008.4631301>]
- [46] J. Brest, A. Zamuda, B. Boskovic, M.S. Maucec, and V. Zumer, "High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction", In: *IEEE Congress on Evolutionary Computation*. IEEE: Hong Kong, 2008, pp. 2032-2039. [<http://dx.doi.org/10.1109/CEC.2008.4631067>]
- [47] L. Tseng, and C. Chen, "Multiple trajectory search for large scale global optimization", In: *Evolutionary Computation*. IEEE: Hong Kong, 2008, pp. 3052-3059.

---

Received: February 12, 2015

Revised: September 28, 2015

Accepted: September 28, 2015

© Xu et al; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution-Non-Commercial 4.0 International Public License (CC BY-NC 4.0) (<https://creativecommons.org/licenses/by-nc/4.0/legalcode>), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.