Comparison Four Different Probability Sampling Methods based on Differential Evolution Algorithm

Lu Qingbo
College of Mechanical Electronic Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China
Email: jh0262@126.com

Zhang Xueliang and Wen Shuhua and Lan Guosheng
College of Mechanical Electronic Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China
Email: zhang_x_l@sina.com

***Abstract*—Differential Evolution (DE) is one kind of evolution algorithm, which based on difference of individuals. DE has exhibited good performance on optimization problem. The current studies almost are based on the simple random sampling method, and so this paper investigates other probability sampling methods, and proposed three novel differential evolution algorithms. The proposed algorithms are compared with the original differential evolution algorithm. The numerical results and Lorenz parameter estimation problem show that the new methods performed better than the original differential evolution algorithm.**

***Index Terms*—simple random sampling, stratified sampling, systematic sampling, cluster sampling, differential evolution, parameter estimation**

I. Introduction

Differential evolution (DE) is a stochastic, population-based optimization method[1, 2], which has been successfully to a wide range of problems as summarized in Price[3]. A number of variations of DE have been developed in the past decade to improve the performance. These researches can be divided two aspects, one is parameter investigation such as the mutation factor and the crossover probability [4-8], and the other is theoretical analyses [9-14]. Tvrdik[4] provided an experimental comparison of two different self-adaptive patterns and influence of exponential crossover. Das et al.[5] provided two new improved variants of DE, DE with random scale factor and DE with time varying scale factor. Liu and Lampinen[6] introduced a new version of the Differential Evolution algorithm with adaptive control parameters – the fuzzy adaptive differential evolution algorithm, which used fuzzy logic controllers to adapt the search parameters for the mutation operation and crossover operation. Teo[7] presented a first attempt at self-adapting the population size parameter in addition to self-adapting crossover and mutation rates. Lu et al.[8] proposed a modified differential evolution by randomly initializing and calculating the scale factor by chaos each generation and introducing a disaster factor into differential evolution algorithm. Omran[9] proposed barebones differential evolution algorithm, BBDE, which is a hybrid algorithm by capitalizing on the strengths of both the barebones PSO and self-adaptive DE strategies. Muelas et al.[10] combined the explorative/exploitative strength of the memetic algorithm and differential evolution algorithm and proposed a hybrid algorithm. Wang et al.[11] presented a novel Differential Evolution (DE) algorithm, called DE enhanced by neighborhood search (DENS), which differs from pervious works of utilizing the neighborhood search in DE, such as DE with neighborhood search (NSDE) and self-adaptive DE with neighborhood search (SaNSDE). Zhang and Sanderson [12] proposed an analytical method to study the evolutionary stochastic properties of the population in differential evolution (DE) for a spherical function and developed the properties of mutation and selection based on which a Gaussian approximate model of DE Zhang and Sanderson[13] proposed a new differential evolution algorithm, JADE, which proposed a new mutation strategy “DE/current-to-pbest” with the optional archive and updated control parameters in an adaptive manner. Zhang et al. [14] proposed a center differential evolution algorithm with adaptive crossover factor; the new algorithm incorporated the center point of the population into the DE algorithm.

Manuscript received August 23, 2011; revised December 5, 2011; accepted December 5, 2011.

Corresponding author, Zhang Xueliang.

doi:10.4304/jait.3.4.206-214

Simple random sampling method is used in all these researches. In this paper, we investigate the other three probability sampling methods and the three novel differential evolution algorithms are presented by applying these sampling methods into differential evolution algorithm. The proposed algorithms are compared with the original differential evolution algorithm. The numerical results and Lorenz parameter estimation problem show that the new methods performed better than the original differential evolution algorithm.

The reminder of the paper is organized as follows: four different probability sampling methods are summarized in Section 2. Section 3 summarized the differential evolution algorithm. The three novel different evolution algorithms are presented in Section 4. Section 5 presents the numerical results and discussions. Parameter estimation for the Lorenz system is investigated in Section 6. Finally, Section 7 concludes the paper.

II. Four different probability sampling methods

Probability sampling is a sampling technique wherein the samples are gathered in a process that gives all the individuals in the population equal chances of being selected. Probability sampling method has four different types, simple random sampling, stratified sampling, systematic sampling and cluster sampling.

A. Simple Random Sampling

Simple random sampling is the easiest form of probability sampling. This sampling method refers to a sampling method that has the following properties.

(a)The population consists of N individuals.

(b)The sample consists of n individuals.

(c)All possible samples of n individuals are equally likely to occur.

One of the best things about simple random sampling is the ease of assembling the sample. It is also considered as a fair way of selecting a sample from a given population since every member is given equal opportunities of being selected.

B. Stratified Sampling

Stratified sampling is a probability sampling technique wherein the researcher divides the entire population into different subgroups (called strata), then randomly selects the final individuals proportionally from the different subgroups.

The strata do not overlap, as shown in figure 1, and they constitute the whole population so that each sampling unit belongs to exactly one stratum.

$$N = N_1 + N_2 + \ldots + N_H \tag{1}$$

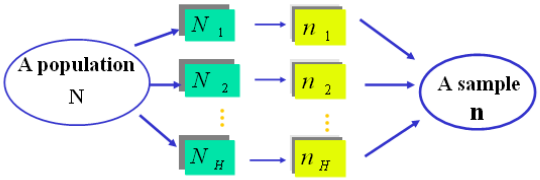


Figure 1. Stratification

If a simple random sample is taken in each stratum, the whole procedure is described as stratified random sampling. Stratified sampling consists of following steps:

(a)The entire population is divided into distinct subpopulations.

(b)Within each stratum, a separate sample is selected.

(c)Separate stratum means (or other statistics) are computed and then properly weighted to form a combined estimate for the entire population.

(d)The variances are computed separately within each stratum and then properly weighted and added into a combined estimate for the population.

C. Systematic Sampling

Systematic sampling is a random sampling technique which is frequently chosen by researchers for its simplicity and its periodic quality. In systematic sampling, the researcher first randomly picks the first item or subject from the population. Then, the researcher will select each n'th subject from the list.

For example, the researcher has a population total of 100 individuals and need 12 subjects. He first picks his starting number 5.Then the researcher picks his interval, 8. The members of his sample will be individuals 5, 13, 21, 29, 37, 45, 53, 61, 69, 77, 85, 97.

D. Cluster Sampling

In cluster sampling, instead of selecting all the subjects from the entire population right off, the researcher takes several steps in gathering his sample population.

First, the researcher selects groups or clusters, and then from each cluster, the researcher selects the individual subjects by either simple random or systematic random sampling. The researcher can even opt to include the entire cluster and not just a subset from it.

For example, a researcher wants to survey academic performance of high school students in China.

(a)He can divide the entire population (population of China) into different clusters (cities).

(b)Then the researcher selects a number of clusters depending on his research through simple or systematic random sampling.

(c)Then, from the selected clusters (randomly selected cities) the researcher can either include all the high school students as subjects or he can select a number of subjects from each cluster through simple or systematic random sampling.

E. Comparison of the Four Probability Sampling Methods

Stratified sampling offers several advantages over simple random sampling. A stratified sample can provide greater precision than a simple random sample of the same size. Because it provides greater precision, a stratified sample often requires a smaller sample, which saves money. We can ensure that we obtain sufficient sample points to support a separate analysis of any subgroup. The main disadvantage of a stratified sample is that it may require more administrative effort than a simple random sample. The main disadvantage of cluster sampling is that cluster sampling generally provides less precision than either simple random sampling or stratified sampling at the sample size. When the increased sample size is sufficient to offset the loss in precision, cluster sampling may be the best choice. Systematic sampling is to be applied only if the given population is logically homogeneous, because systematic sample units are uniformly distributed over the population.

III. Differential Evolution Algorithm

Differential evolution (DE) is an evolutionary algorithm proposed by Storn and Price. The basic DE algorithm is described in detail below with reference to the four key operators: initialization, mutation, crossover and selection.

Initialization: Before the population can be initialized, both upper and lower bounds for each parameter must be specified. Once initialization bounds have been specified, a random number generator assigns each parameter of every vector a value from within the prescribed range. For example, the initial value (g=0) of the jth parameter of the ith vector is

$$x_{\mathrm{i,j}}(0) = \mathrm{rand}(0,1)(b_{\mathrm{U,j}} - b_{\mathrm{L,j}}) + b_{\mathrm{L,j}} \tag{2}$$

The random number generator, rand (0, 1), returns a uniformly distributed random number from within the range [0,1].

Mutation: Once initialized, DE mutates and recombines the population to produce a population of NP trial vectors. For each parent, $x_i(t)$, of generation t, a trail vector, $u_i(t)$, is created by mutating a target vector. The target vector, $x_{r3}(t)$, is randomly selected, with $i \neq r_3$. Then, two individuals $x_{r1}(t)$, and are randomly selected with $i \neq r_2 \neq r_1$, and the difference vector, $x_{r1}(t) - x_{r2}(t)$, is calculated. The trail vector is then calculated as

$$u_i(t) = x_{r3}(t) + F\left(x_{r1}(t) - x_{r2}(t)\right) \tag{3}$$

where the last term represents the mutation step size. In the above, F is a scale factor used to control the amplification of the differential variation. Note that $F \in (0,2)$.

Crossover: DE follows a discrete recombination approach where elements from the vector, $x_i(t)$, are combined with elements from the trail vector, $u_i(t)$, to produce the offspring, $v_i(t)$. Using the binomial crossover,

$$v_{i,j}(t) = \begin{cases} u_{i,j}(t) & \mathit{if\ rand(0,1) < Cr\ or\ j = r} \\ x_{i,j}(t) & \mathit{otherwise} \end{cases}$$

(4)

where $j = 1,2,...,D$ refers to a specific dimension. r= rand(0, D). In the above, Cr is the probability of reproduction (with $Cr \in [0,1]$).

Selection: DE evolution implements a very simple selection procedure. The generated offspring, $v_i(t)$, replaces the parent, $x_i(t)$, only if the fitness of the offspring is better than that of the parent.

Storn and Price also proposed ten different strategies for DE based on the individual being perturbed, the number of individuals used in the mutation process and the type of crossover used. The strategy described above is known as DE/rand/1, meaning that the target vector is randomly selected, and only one difference vector is used. This strategy is considered to be the most widely used and it is the one used in this paper. Other main strategies include DE/best/1, DE/best/2, and DE/rand-to-best/1. The notation, DE/*x*/*y*, is used where *x* represents the individual being perturbed and *y* is the number of difference vectors used to perturb *x*.

IV. Three Novel Differential Evolution Algorithm

In original differential evolution algorithm, the method used to generate trail vector and target vector is simple random sampling method. The mainly different between the three novel differential evolution and DE is that the method used in the three novel differential evolution algorithms to generate trail vector and target vector are not randomly selected, but stratified sampling method, systematic sampling method and cluster sampling method.

A.Stratified Sampling Differential Evolution Algorithm

The stratified sampling differential evolution algorithm (SSDE) uses the stratified sampling method to generate the trail vector and target vector. In order to use stratified random sampling method, a quick sorting method is employed to sort the population by fitness. NP represents the population size. The pseudo code of the stratified sampling process is listed in Algorithm 1.

Algorithm 1 The pseudo code of the stratified sampling process

```
getIndex (int* r1,int* r2,int* r3,int i){
 Quick sort population by fitness;
 //select the layer index
 int k1,k2,k3=rand(1,3) and k1≠k2≠k3;
   if(k1==1 and k2==2 and k3==3)
     *r1=rand(0,[NP/3]),* r2=rand([NP/3], [2NP/3]),
     *r3=rand([NP/3],NP]) and *r1 or *r2 or *r3≠i;
   if(k1==1 and k2==3 and k3==2)
     *r1=rand(0,[NP/3]), *r2=rand([NP/3],NP]) ,
     * r3=rand([NP/3], [2NP/3]) and *r1 or *r2 or *r3≠i;
   if(k1==2 and k2==1 and k3==3)
     * r1=rand([NP/3], [2NP/3]),*r2=rand(0,[NP/3]),
     *r3=rand([NP/3],NP]) and *r1 or *r2 or *r3≠i;
   if(k1==2 and k2==3 and k3==1)
     * r1=rand([NP/3], [2NP/3]), *r2=rand([NP/3],NP]) ,
     *r3=rand(0,[NP/3]) and *r1 or *r2 or *r3≠i;
   if(k1==3 and k2==1 and k3==2)
     *r1=rand([NP/3],NP]), *r2=rand(0,[NP/3]),
     * r3=rand([NP/3], [2NP/3]) and *r1 or *r2 or *r3≠i;
```

```
    if(k1==3 and k2==2 and k3==1)
   *r1=rand([NP/3],NP]) , * r2=rand([NP/3], [2NP/3]),
   *r3=rand(0,[NP/3]) and *r1 or *r2 or *r3≠i;
   }
```

B.Systematic Sampling Differential Evolution Algorithm

The systematic sampling differential evolution algorithm (SYSDE) uses the systematic sampling method to generate the trail vector and target vector. The pseudo code of the systematic sampling process is listed in Algorithm 2.

Algorithm 2 The pseudo code of the systematic sampling process

```
getIndex (int* r1,int* r2,int* r3,int i){
    *r1=rand(0,[(NP-NP%3)/3]) and *r1≠i;
    *r2=*r1+(NP-NP%3)/3;
    *r3=*r2+(NP-NP%3)/3;
}
```

C.Cluster Sampling Differential Evolution Algorithm

The cluster sampling differential evolution algorithm (CDE) uses the cluster sampling method to generate the trail vector and target vector. The pseudo code of the cluster sampling process is listed in Algorithm 3.

Algorithm 3 The pseudo code of the cluster sampling process

```
getIndex (int* r1,int* r2,int* r3,int i, int NC ){
//randomly select the subgroup index.
t=rand(0,NC);
//calculate the number of individual in each subgroup.
count=(NP-NP%NC)/NC;
//calculate the start number of the selected subgroup.
startNum=(NP-NP%N)/N*temp;
 endNum=startNum+count;
 *r1=rand(startNum,endNum), and *r1≠i;
*r2=rand(startNum,endNum), and *r2≠*r1≠i;
*r3=rand(startNum,endNum), and *r3≠*r1≠*r2≠i;
}
```

V. Experimental results

This section compares the performance of the SSDE algorithm, SYSDE algorithm and CDE algorithm with the original differential evolution algorithm (SDE).

The following functions have been used to compare the performance of SSDE, SYSDE and CDE with SDE. These benchmark functions provide a balance of unimodal and multimodal functions.

A .Sphere function, defined as

$$f(\boldsymbol{x})=\sum_{i=1}^{D}x_i^2$$

where $\boldsymbol{x}^*=(0,0,...,0), f(\boldsymbol{x}^*)=0$

for $x_i \in [-100,100]$.

B. Schwefel' s problem 2.22, defined as

$$f(\boldsymbol{x})=\sum_{i=1}^{D}|x_i|+\prod_{i=1}^{D}|x_i|$$

where $\boldsymbol{x}^*=(0,0,...,0), f(\boldsymbol{x}^*)=0$

for $x_i \in [-10,10]$.

C. Step function, defined as

$$f(\boldsymbol{x})=\sum_{j=1}^{D}\left(\lfloor x_i+0.5\rfloor\right)^2$$

where $\boldsymbol{x}^*=(0,0,...,0), f(\boldsymbol{x}^*)=0$

for $x_i \in [-100,100]$.

D. Rosenbrock function, defined as

$$f(\boldsymbol{x})=\sum_{j=1}^{D-1}\left[100\left(x_j^2-x_{j+1}\right)^2+\left(x_j-1\right)^2\right]$$

where $\boldsymbol{x}^*=(1,1,...,1), f(\boldsymbol{x}^*)=0$

for $x_i \in [-30,30]$.

E. Rotated hyper-ellipsoid function, defined as

$$f(\boldsymbol{x})=\sum_{i=1}^{D}\left(\sum_{j=1}^{i}x_j\right)^2$$

where $\boldsymbol{x}^*=(0,0,...,0), f(\boldsymbol{x}^*)=0$

for $x_i \in [-100,100]$.

F. Generalized Swefel's problem 2.26, defined as

$$f(\boldsymbol{x})=\sum_{j=1}^{D}-x_i sin\left(\sqrt{|x_i|}\right)$$

where $\boldsymbol{x}^*=(420.9678,420.9678,...,420.9678)$

$f(\boldsymbol{x}^*)=-12569.5$ for $x_i \in [-500,500]$.

G. Rastrigin function, defined as

$$f(\boldsymbol{x})=\sum_{j=1}^{D}\left[x_i^2-10cos(2\pi x_i)+10\right]$$

where $\boldsymbol{x}^*=(0,0,...,0), f(\boldsymbol{x}^*)=0$

for $x_i \in [-5.12,5.12]$.

H. Ackley's function, defined as

$$f(\boldsymbol{x})=-20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)$$
$$-exp\left(\frac{1}{D}\sum_{i=1}^{D}cos(2\pi x_i)\right)+20+e$$

where $\boldsymbol{x}^*=(0,0,...,0), f(\boldsymbol{x}^*)=0$

for $x_i \in [-32,32]$.

I. Griewank function, defined as

$$f(\boldsymbol{x})=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}cos\left(\frac{x_i}{\sqrt{i}}\right)+1$$

where $\boldsymbol{x}^* = (0,0,...,0), f(\boldsymbol{x}^*) = 0$

for $x_i \in [-600,600]$.

J. Six-hump Camel-back function, defined as

$$f(\boldsymbol{x}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

where $\boldsymbol{x}^* = (-0.08983, 0.7126)$

$f(\boldsymbol{x}^*) = -1.0316285$ for $x_i \in [-5,5]$.

Sphere, Schwefel's problem 2.22, Rosenbrock and Rotated hyper-ellipsoid are unimodal, while the Step function is a discontinuous unimodal function, Schwefel's problem 2.26, Rastrigin, Ackley and Griewank are difficult multimodal functions where the number of local optima increase exponentially with the problem dimension. The Camel-back function is a low-dimensional function with only a few local optima.

For all the algorithms used in this section, the population size NP set 100. All functions were implemented in 30 dimensions except for the two-dimensional Camel-back function. The results reported in this section are averages and standard deviations over 50 simulations. Each simulation was allowed to run for 50,000 evaluations of the objective function. *F*=0.5 and *Cr*=0.1, used DE/rand/1/bin strategy.

Table 1 summarizes the results obtained by applying the different approaches to all benchmark functions. The results show that the CDE, SSDE and SYSDE performed better than DE. The SSDE performed best than other three approaches. Fig.2 illustrates results for the selected benchmark functions. For the Sphere function, Fig.2a shows that SSDE achieved a faster reduction in fitness than SYSDE, CDE and SDE. For the Rosenbrock function, Fig.2b shows that SSDE achieved a faster reduction in fitness than SYSDE, CDE and SDE, and reached a good solution faster than the other approaches. For the Rotated hyper-ellipsoid function, Fig.2c shows that SSDE reached a good solution than other algorithms. For the Schwefel problem 2.26 function, Fig.2d shows that SSDE reached a good solution faster than the other approaches. For the Rastrigin function, Fig.2e shows that SDE achieved a faster reduction in fitness than the other approaches, but SSDE and CDE obtained a good solution than the other approaches. For the Ackley function, Fig.2f shows that SSDE achieved a faster reduction in fitness than the other approaches.

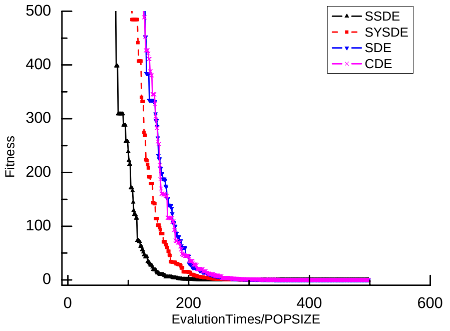
TABLE I.
MEAN AND STANDARD DEVIATION OF THE BEST-OF-RUN SOLUTION FOR 50 RUNS

Function	SDE	SYSDE	CDE	SSDE
Sphere	0.001157 (0.000253)	0.000072 (0.000018)	0.000725 (0.000324)	**0.00000 (0.00000)**
Schwefel Problem2.22	0.00491 (0.00055)	0.00129 (0.00017)	0.00367 (0.00074)	**0.00004 (0.00001)**
Step	0.00000 (0.00000)	0.00000 (0.00000)	0.00000 (0.00000)	0.00000 (0.00000)
Rosenbrock	151.788 (23.856)	73.267 (19.6488)	144.566 (36.9609)	**40.644227 (21.0556)**
Rotated hyper-ellipsoid	19786.739 (2888.44)	18660.5212 (2485.94)	19789.3976 (3230.92)	**15844.9244 (2219.54)**
Schwefel problem 2.26	11509.472 (312.0277)	-12315.330 (206.56534)	-12168.749 (256.6817)	**-12569.483 (0.00364)**
Rastrign	40.760631 (4.099100)	37.562365 (3.180995)	36.836295 (4.488318)	**5.76638 (3.06823)**
Ackley	0.009931 (0.001264)	0.002387 (0.000298)	0.007937 (0.001949)	**0.00011 (0.00002)**
Griewank	0.009259 (0.00274)	0.001280 (0.00097)	0.006949 (0.0048)	**0.00001 (0.000001)**
SixJump	-1.031628 (0.000000)	-1.031628 (0.000000)	-1.031628 (0.000000)	-1.031628 (0.000000)

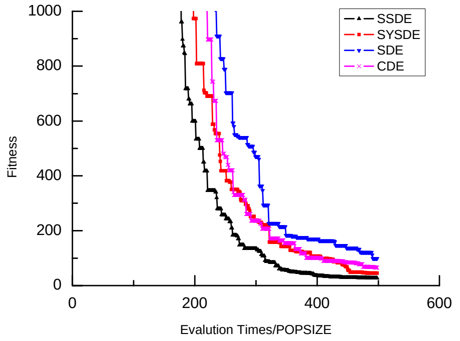
Figure.3 illustrates diversity for selected benchmark functions. Diversity has been calculated using

$$diversity = \frac{1}{POPSIZE}\sum_{i=1}^{POPSIZE}\sqrt{\sum_{j=1}^{D}\left(x_{ij}(t) - \bar{x}_j(t)\right)^2}$$

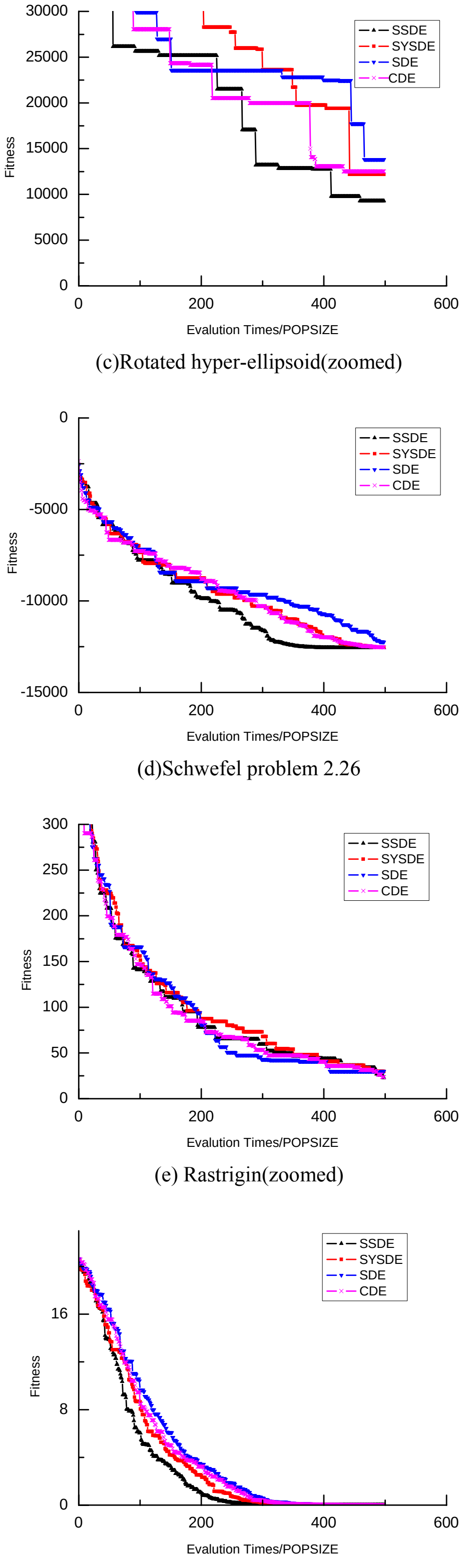
where $\bar{x}_j$ is the average of the *j*th dimension over all individuals, i.e. $\bar{x}_j(t) = \frac{1}{POPSIZE}\sum_{i=1}^{t} x_{ij}(t)$.



(a)Sphere(zoomed)



(b)Rosenbrock(zoomed)



(c)Rotated hyper-ellipsoid(zoomed)

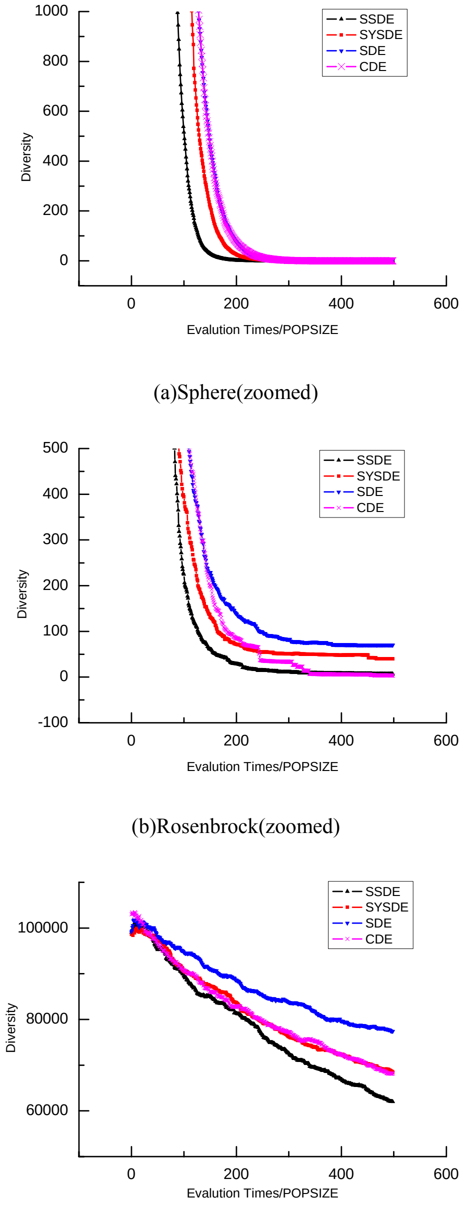
(d)Schwefel problem 2.26

(e) Rastrigin(zoomed)

(f) Ackley

Figure 2. Performance comparison of the different methods for selected benchmark functions

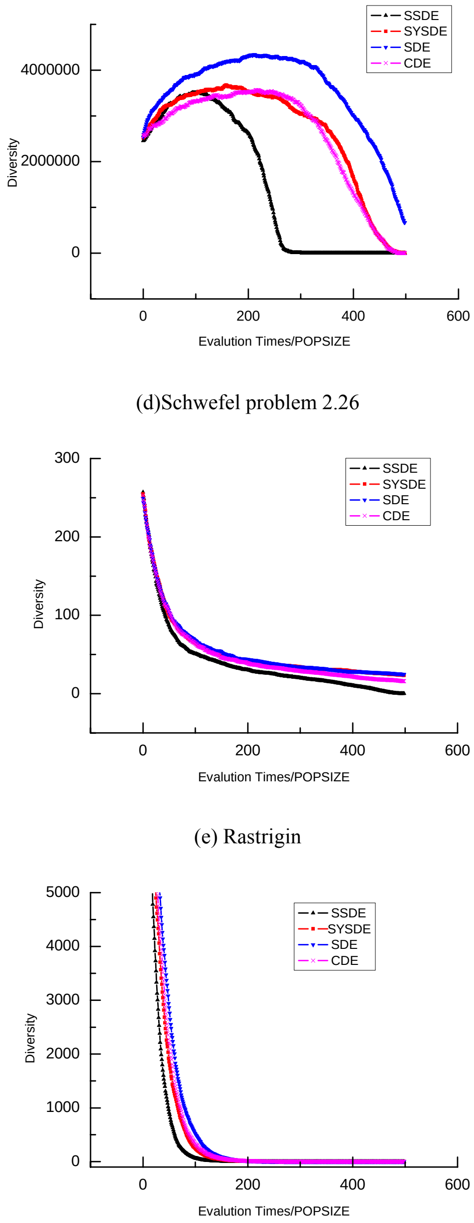
For the Sphere function, Fig.3a shows that the SSDE exhibited the fasted reduction in diversity enabling it to converge faster the other approaches. For the Rosenbrock function, Fig.3b shows that the SSDE exhibited the fasted reduction in diversity enabling to converge faster than the other approaches. The SDE was the slowest reduction in diversity which might cause its slow convergence. For the Rotated hyper-ellipsoid function, Fig.3c shows that SSDE achieved a faster reduction in diversity than other algorithms. For the Schwefel problem 2.26 function, Fig.3d shows that diversity increased firstly and then decreased, and SSDE exhibited the fasted reduction in diversity enabling to converge faster than the other approaches. For the Rastrigin function and Ackley, Fig.3e and Fig.3f show that SSDE exhibited the fasted reduction in diversity enabling to converge faster than the other approaches.



(a)Sphere(zoomed)

(b)Rosenbrock(zoomed)

(c)Rotated hyper-ellipsoid



(d)Schwefel problem 2.26

(e) Rastrigin

(f) Ackley(zoomed)

Figure 3. Diversity comparison of the different methods for selected benchmark functions

VI. Parameter Estimation in the Lorenz Model

The Lorenz model has been widely used for studies involving prediction and data assimilation in chaotic systems. The model consists of three variables x, y and z, which evolve according to the equations

$$x' = a(y - x) \tag{5}$$

$$y' = bx - y - xz \tag{6}$$

$$z' = xy - cz \tag{7}$$

where a,b,c are three constant parameters that are generally given the values 10, 28 and 8/3, respectively. For these values, the model variables follow a highly chaotic orbit. All calculations in this paper were performed using a fourth-order Runge-Kutta method with a time-step of 0.01.

The optimization model is established as follows:

$$\min \varepsilon = \frac{1}{M}\sum_{k=1}^{M}\left\| X_k - Y_k \right\|^2 \tag{8}$$

where M is the sequence length of state variable, X_k(k=1,2,…,M) is the kth state variables sequence at the true value of parameters of chaotic system, and Y_k (k=1,2,…,M) is the kth state variables sequence at the estimated value of parameters of chaotic system. For the Lorenz chaotic system, a, b, c are the decision variable.

For all the algorithms used in this section, the population size NP set 60. The results reported in this section are averages and standard deviations over 20 simulations. Each simulation was allowed to run for 30,000 evaluations of the objective function. F=0.5 and Cr=0.1, used DE/rand/1/bin strategy.

Table 2 summarizes the best solution obtained by applying different approaches. The results show that the SSDE obtained the best one of the four solutions. Table 3 summarizes the statistical results obtained by applying different approaches. The results show that SSDE performed better than the other methods in all means and standard deviation of parameters and best fitness.

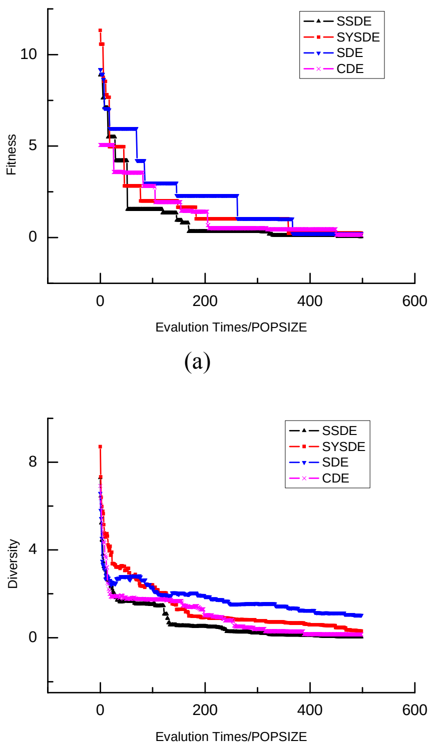
TABLE II.
Comparison of the Best Solution by Different Methods

	SDE	SYSDE	CDE	SSDE
a	10.022700	9.926893	10.004860	10.002669
b	27.991236	28.062918	27.997970	27.993636
c	2.668656	2.667592	2.667006	2.665797
ε	0.219830	0.239681	0.149356	0.045487

TABLE III.
Statistical Results of Different Methods

	SDE	SYSDE	CDE	SSDE
a	10.06818 (0.23197)	10.03677 (0.17154)	10.0591 (0.21308)	10.01558 (0.03012)
b	27.93083 (0.19324)	27.95608 (0.14123)	27.94265 (0.20195)	27.98527 (0.03098)
c	2.66419 (0.00919)	2.66397 (0.00672)	2.66514 (0.01321)	2.66622 (0.00232)
ε	0.910464 (0.385202)	0.747560 (0.255342)	0.955321 (0.820300)	0.150204 (0.086611)

Figure 4 illustrates performance comparison of the different methods for Lorenz parameter estimation problem. Figure 4a shows that SSDE achieved a faster reduction in fitness and reached a good solution than the other approaches. Figure 4b shows that SSDE exhibited the fasted reduction in diversity enabling to converge faster than the other approaches.



(a)

(b)

Figure 4. Performance comparison of the different methods for Lorenz parameter estimation problem

VII. Conclusions

This paper presented three different models for differential evolution algorithm by investigating three probability sampling method. These approaches were compared with the simple random sampling method which used in the original differential evolution algorithm. The results show that these methods performed better than the original method in all selected benchmark functions. The results also show that the SSDE performed better than the SYSDE and CDE. This paper also investigated the parameter estimation problem, compared the results obtained by the proposed three algorithms and original differential evolution algorithm in the Lorenz parameter estimation problem. The results show that the SSDE performed best than the other strategies.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No. 50775153), Ph.D. Program Foundation of Ministry of Education of China (Grant No. 20091415110002).

References

[1] Storn, R. and K. Price, “Differential Evolution-A simple and efficient adaptive scheme for global optimization over continuous spaces,” *ICSI Technical Report TR-95-012,* March 1995.

[2] Storn, R. and K. Price, “Differential Evolution-a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 4, pp. 359-431, November 1997.

[3] Price, K. and R. Storn, “Differential Evolution: A Practical Approach to Global Optimization,” *Springer*, 2005.

[4] Tvrdik, J., “Adaptive differential evolution and exponential crossover,” *International Multiconference on Computer Science and Information Technology,* Wisia, 2008, pp.927-931.

[5] Das, S., A. Konar, and U.K. Chakraborty, “Two improved differential evolution schemes for faster global search,” *Genetic and Evolutionary Computation Conference*, Washington, DC,2005, pp.991-998.

[6] Liu, J. and J. Lampinen, “A fuzzy adaptive differential evolution algorithm ,” *Soft Computing - A Fusion of Foundations, Methodologies and Applications,* vol. 6, pp.448-462, September 2005.

[7] Teo, J., “Exploring dynamic self-adaptive populations in differential algorithm,” *Soft Computing - A Fusion of Foundations, Methodologies and Applications,* vol. 8, pp.673-686, October 2006.

[8] Lu Qingbo, Zhang Xueliang, Wen Shuhua, et al., “Modified Differential Evolution and Its Application,” *Transactions of the Chinese Society for Agricultural Machinery,*vol. 2, pp.193-197, February 2010.

[9] Omran, M.G.H., A.P. Engelbrecht, and A. Salman, “Bare Bones Differential Evolution,” *European Journal of Operational Research,*vol.1, pp.128-139, July 2009.

[10] Muelas, S., A. La Torre, and J. Pea, “A Memetic Differential Evolution Algorithm for Continuous Optimization,” *2009 Ninth International Conference on Intelligent Systems Design and Applications.* Pisa, 2009, pp. 1080 – 1084.

[11] Wang, H., Z. Wu, and S. Rahnamayan, “Differential Evolution enhanced by neighborhood search,” *IEEE Congress on Evolutionary Computation.* Barcelona, 2010,pp. 1-8.

[12] Zhang, J. and A.C. Sanderson, “An approximate Gaussian model of differential evolution with spherical fitness functions,” *IEEE Congress on Evolutionary Computation.* Singapore, 2007. pp. 2220-2228.

[13] Zhang, J. and A.C. Sanderson, “JADE: Adaptive Differential Evolution With Optional External Archive,” *IEEE Congress on Evolutionary Computation.* NY,USA, 2009. pp. 945-958.

[14] Zhang Xueliang,Lu Qingbo, Wen Shuhua, et al., “Modified Differential Evolution for Constrained Optimization and Its Application” *Transactions of the Chinese Society for Agricultural Machinery,*vol. 8, pp.135-139, August 2008.

Lu Qingbo graduated from Northeastern University, China in 2000, and received the master of science degree from TaiYuan University of Science and Technology , China, in 2008. He is currently a Ph.D. candidate in TaiYuan University of Science and Technology ,China.

His research interests include mechanical designing and optimization.

Zhang Xueliang graduated from TaiYuan University of Technology, China in 1984, and received the Ph.D degree from Xi'an University of Technology, China, in 1998. He is currently a professor in TaiYuan University of Science and Technology ,China.

His research interests include mechanical designing and optimization and mechanical structural dynamics .

Wen Shuhua graduated from ShenYang University of Technology, China in 1987. She is currently a professor in TaiYuan University of Science and Technology ,China.

Her research interests include mechanical structural dynamics and optimization algorithm.

Lan Guosheng graduated from TaiYuan University of Technology, China in 2000, and received the master of science degree from TaiYuan University of Science and Technology , China, in 2005. He is currently a Ph.D. candidate in TaiYuan University of Science and Technology ,China.

His research interests include mechanical structural dynamics and optimization.