2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Differential Evolution with Replacement Strategy for Real-Parameter Numerical Optimization

Changjian Xu, Han Huang, Shujin Ye*
School of Software Engineering
South China University of Technology
Guangzhou, China
e-mail: x.changjian@mail.scut.edu.cn, hhan@scut.edu.cn, yeshujin2@163.com

***Abstract*—Differential Evolution (DE) has been widely used as a continuous optimization technique for several problems like electromagnetic optimization, bioprocess system optimization and so on. However, during the optimization process, DE's population may stagnate local optima where the algorithm has to spend a large number of function evaluations to get rid of them. This paper presents an improved DE algorithm (denoted as RSDE) which combines two Replacement Strategies (RS).The motivation of RS is that replacing an unimproved individual and replacing a premature population using RS which can enhance the DE exploitation performance and exploration performance respectively. We tested the RSDE performance using the newly Single Objective Real-Parameter Numerical Optimization problems provided by the CEC 2014 Special Session and Competition. Moreover, computational results, convergence figures and the performance of these two RS will be presented to discuss the feature of RSDE.**

Keywords—Differential Evolution; Replacement Strategy; Single Objective Real-Parameter Numerical Optimization problem

I. INTRODUCTION

Differential evolution (DE) is a popular optimization method proposed by Storn and Price in 1997 [1]. The continuous optimization algorithm DE is similar to the Genetic Algorithm (GA) which has been considered as the robust solution of discrete optimization problems. However, the major difference between DE and the GA is that DE mutates its population vectors before crossover operation while GA crosses the Genomes after picking up the parent vectors from the population vectors and finally mutates the offspring.

DE consists of mutation process, crossover process and the selection process. In the mutation process, DE may pick up the random individuals or the best individual to product an offset vector which will be added to the target vector to product the mutant vector. After mutation process, crossover combines the mutant vectors with the target vectors so that the trial vectors can be generated. Finally, selection process, which is based on the vector function fitness, picks up the better vectors to be the next parents between the target vector and the trial vector.

DE plays an important role in the several fields of engineering optimization [2-4] because of the advantage of DE, such as robustness and effectiveness. Some examples of the usage of DE to optimize the engineering problems are given as follows. DE was added to optimal control problems of a bioprocess system [2] by Chiou and Wang. Furthermore, a DE scheme was used to train the neural networks [3]. After that, a framework which is based on DE algorithm was designed for the electromagnetic optimization [4]. Therefore, the improvement of DE can make great a difference to several engineering fields.

Recently, more and more researchers of meta-heuristic algorithm have paid their attention on the improvement of the DE performance. After Storn and Prince provided the DE algorithm, Qin and Suganthan introduced a Self-adaptive DE (SaDE) [6] to improve the original DE by using a self-adaptive parameter strategy. With such a adaptive parameter (crossover rate and scale factor) strategy, SaDE show its robustness in learning the evolution information to solve a new problem without user pre-defined parameter. Moreover, Zhang and Sanderson presented a novel DE algorithm[7] with a 'DE/current-to-*p*bset' mutation strategy which is a generalization of 'DE/current-to-best'[8]. This *p*best mutation strategy maintain the diversity of population well and provide a progress direction on the basis of the historical data. However, little research has been carried out on local optimum detection and recovery of the DE algorithm. The motivation of our study is to design two replacement strategies to ameliorate the unimproved individuals and improve the algorithm performance while DE's population loses in the local optimum respectively.

The aim of the paper is to carry out a performance study of RSDE on CEC 2014 benchmark functions. The rest of the paper is organized as follow. Original DE algorithm is introduced in Section II. The idea of Replacement Strategy and computational procedure of RSDE are presented in Section III. The result discussion is illustrated in the Section IV. Finally, Section V concludes the paper.

II. DIFFERENTTIAL EVOLUTION

The original DE algorithm is presented in detail as follow: Let S be a finite subspace of the n-dimensional real domain R^n, and let $f: S \to R$ be an n-dimensional real function. DE provides evolution for a population of NP n-dimensional

The author marked with character * is the Corresponding author individual vectors $X_i^t = (x_{i,1}^t,..,x_{i,D}^t) \in S, i = 1,..,NP$, where D is the objective function dimension and t is denoted as the iteration number of the DE algorithm. The initial population should be randomly distributed between the upper bound $X_{\max} = (x_{\max},..,x_{\max})$ and lower parameter bound $X_{\min} = (x_{\min},..,x_{\min})$. After the initialization, mutation process, crossover process and selection process will be operated from one iteration to next iteration until some specific stopping criteria are satisfied. According to [5], the three processes of DE can be summarized as follows:

1) Mutation:

$$v_i^{t+1} = X_i^t + F \cdot (X_{r_1}^t - X_{r_2}^t) \quad (1)$$

Where $i, r_1, r_2 \in [1, NP]$ are random and mutually different integers, $X_{r_1}^t, X_{r_2}^t$ are different from the target vector. Scale factor F > 0 is a real constant factor and is often set to 0.5.

2) Crossover

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1} \; if \; rand_j \le CR \; or \; j = j_{rand} \\ x_{i,j}^t \quad otherwise \end{cases}$$

Where CR is a crossover constant in $[0,1)$, $j \in [1, D]$, $rand_j$ is a uniformly distributed random number and j_{rand} is a random integer from 1 to D, which guarantee that at least one dimension parameter is different from the target vector X_i^t .Here, vector u_i^{t+1} is considered as the trial vector.

3) Selection

The selection process is based on the vector function fitness value. The algorithm compares the target vector fitness value $f(X_i^t)$ and the trial vector function value $f(u_i^{t+1})$. If $f(u_i^{t+1}) < f(X_i^t)$, the trial vector u_i^{t+1} will be chosen for the next generation. Otherwise, X_i^t will be chosen.There are also several different mutation strategies of DE [5]:

"DE/rand/1" [1]:

$$v_i^{t+1} = X_{r_1}^t + F \cdot (X_{r_2}^t - X_{r_3}^t) \quad (2)$$

"DE/best/1" [5]:

$$v_i^{t+1} = X_{best}^t + F \cdot (X_{r_1}^t - X_{r_2}^t) \quad (3)$$

"DE/current to best/1" [8]:

$$v_i^{t+1} = X_i^t + F \cdot (X_{best}^t - X_i^t) + F \cdot (X_{r_1}^t - X_{r_2}^t) \quad (4)$$

"DE/best/2" [9]:

$$v_i^{t+1} = X_{best}^t + F \cdot (X_{r1}^t - X_{r2}^t) + F \cdot (X_{r3}^t - X_{r4}^t) \quad (5)$$

"DE/rand/2" [6]:

$$v_i^{t+1} = X_{r1}^t + F \cdot (X_{r2}^t - X_{r3}^t) + F \cdot (X_{r4}^t - X_{r5}^t) \quad (6)$$

Where $r1, r2, r3, r4, r5$ are random and mutually different integers picked up in the range $[1, NP]$.

III. DIFFERENTIAL EVOLUTION WITH REPLACEMENT STRATEGY

A. Idea of Replacement Strategy

Much research, such as self-adaptive parameter optimization and mutation strategies, has been devoted to improve the DE algorithm performance. However, little research has been carried out on the strategy that would help DE population get rid of the local optimum. Since local optimums are the traps of the objective function, a large number of function evaluations (FEs) should be used so that the population would have the opportunity to get rid of local optimums. To improve DE performance and reduce the waste of the FEs while the population stagnate the local optimum, Replacement Strategy (RS) is presented in this paper.

Firstly, an individual replacement strategy is added to original DE algorithm. The aim of DE algorithm is to evolve the population vectors and find out the best vector parameter X^* of the objective function. Individual replacement strategy will verify whether the individual has been improved or not. If the individual vector has not been improved for several iterations (denote as α iterations), it will be replaced by a vector which combines the best individual vector with a random vector parameter offset. With the individual replacement strategy, the convergence speed of DE can be accelerated.

Secondly, a population replacement strategy is used to avoid the population waste a lot of FEs to search a narrow area where DE may not find the best vector parameter of the objective function. Every β iterations, the population replacement strategy will perform once to compare the function value of best vector in current population with the function value of the best in last population before β iterations. If the current best fitness has not been improved, the population will be regenerated using the best vector of current population. Otherwise, the last best vector will be replaced by the best vector of current population and the population will be kept and evolved in the next generation. If the population replacement strategy has been successfully performed for γ times, it can be considered that the population has fallen into an unrecoverable situation so RSDE will regenerate the population using the uniform distribution to cover the entire parameter space randomly.

We introduce the above replacement strategy into the original DE algorithm and develop a new DE algorithm with replacement strategy (RSDE). DE exploitation performance and the exploration performance are enhanced by the RS. The population has more opportunities to get rid of the local optimum and to search for the best vector parameter of the objective function. In the next subsection, we will describe the computational procedure of RSDE.

B. Computational procedure of RSDE

The RSDE uses the DE/current-to-best/1 as the mutation strategy. The computational procedure of the RSDE pseudo-code is presented as follow.

Algorithm 1: DE/best to current/1/bin with RS
{ parameter definition ***}***
{ P^t .. population after t generations}
{NP .. population size}
{D .. dimension of the objective function}
{ $x_{\min}$ .. lower bound of the search range}
{ $x_{\max}$ .. upper bound of the search range}

{ $rand_u(0,1)$ .. uniformly distributed random number}
{Gauss(0,1) .. Gauss distribute random number}
{ M_i .. counter for individual replacement strategy}
{ X_{hbest} .. the best vector so far from the DE begin}
{ X_{best}^{last} .. the last best vector of population}

{ RSDE computational procedure ***}***

Initialize the generation counter $t = 0$
Initialize the population set $P^t = \{X_1^t, .., X_{NP}^t\}$ where $X_i^t = (x_{i,1}^t, .., x_{i,D}^t)$
Initialize the population
Evaluate the fitness value of the individual in P^t respectively. Denoted as $f(X_i^t)$
while the specific stopping criteria is not satisfied **do**
  for $i = 1 \to NP$
    { individual replacement strategy ***}***
    if $M_i < \alpha$ **then**
      $v_i^{t+1} \leftarrow X_i^t + F \cdot (X_{best}^t - X_i^t) + F \cdot (X_{r1}^t - X_{r2}^t)$
      for $j = 1 \to D$
        if $CR \ge rand_u$ **or** $j = j_{rand}$ **then**
          $u_{i,j}^{t+1} \leftarrow v_{i,j}^{t+1}$
        else
          $u_{i,j}^{t+1} \leftarrow x_{i,j}^t$
        end if
      end for
      Evaluate the fitness value of u_i^{t+1}
      if $f(u_i^{t+1}) < f(X_i^t)$ **then**
        $X_i^{t+1} \leftarrow u_i^{t+1}, M_i \leftarrow 0$
      else
        $X_i^{t+1} \leftarrow X_i^t, M_i \leftarrow M_i + 1$
      end if
    else {** if $M_i > \alpha$ **}
      $X_i^{t+1} \leftarrow X_{hbest}, j \leftarrow j_{rand}$
      $X_{i,j_{rand}}^{t+1} = X_{i,j_{rand}}^{t+1} + Gauss(0,1) \cdot (x_{\max} - x_{\min}) \cdot l_t$
end for{ ** terminal of **for** $i = 1 \to NP$ **}
{ population replacement strategy***}***
if mod(t , β) == 0 **then**
  if $\frac{f(X_{best}^t) - f(X_{best}^{last})}{f(X_{best}^{last})} < \varepsilon$ **then**
    $count \leftarrow count + 1$
  end if
  if count $< \gamma$ **then**
    for $i = 1 \to NP$
      for $j = 1 \to D$
        $x_{i,j}^t \leftarrow Gauss(0,1) \cdot \frac{x_{\max} - x_{\min}}{\delta} + x_{best,j}^t$
      end for
      $M_i \leftarrow 0$
    end for
  else
    for $i = 1 \to NP$
      for $i = 1 \to NP$
        $x_{i,j}^{t+1} \leftarrow rand_u(0,1) \cdot (x_{\max} - x_{\min}) + x_{\min}$
      end for
    end for
    $count \leftarrow 0$
  end if
end if
  $X_{best}^{last} \leftarrow X_{best}^t$
end while

C. Replacement Strategy for Individual vector

Although DE algorithm has rapid convergence speed, there are still a few individuals can not be improved for several iterations. In this situation, the unimproved individual vector can be considered that it has stagnated local optima and it would waste a large number of FEs to escape from the local optima. The first replacement strategy in RS is aim to solve this problem. Compared with the function value in last interaction, if the function value is not be improved, the corresponding individual can be detect that it has not been improve for one interaction. Once an individual is detected that it has not been improved for α interactions, it will be replaced by the so far best individual with one dimension mutated using Gauss distributed random offset which is relative to the current interaction. Equation (6) and Equation (7) presented as follow shows the key idea of the individual replacement strategy.

$$X_j^{t+1} = X_{hbest} \quad (6)$$

$$X_{i,j_{rand}}^{t+1} = X_{i,j_{rand}}^{t+1} + Gauss(0,1) \cdot (x_{\max} - x_{\min}) \cdot l_t \quad (7)$$

Where X_{hbest} is the so far individual, the X_i^{t+1} will replace the unimproved individual for next interaction, $Gauss(0,1)$ is a Gauss random number with expectation 0 and 1 standard deviation 1, the max bound of function domain $x_{\max}$ and the min bound of function domain $x_{\min}$. What's more, l_t in (7) positively correlates to the interaction reflected by the function evaluations FEs. And the equation of l_t is present as follow.

$$l_t = (\frac{FE}{10^5})^3 \quad (8)$$

Since the l_t increases in every interaction, the offset weight will increase, which will provide a small offset in early period for better relocation and a large offset in last period for increasing the population diversity. Using the information provided by the so far individual will lead the unimproved individual to accelerate the convergence speed.

D. Replacement Strategy for population vectors

Subsection C introduces the RS for individual vector which is considered as an acceleration of convergence speed and a protection from FEs waste because the unimproved individual need more than α interaction to get rid of the local optima. And this subsection will present the other type of RS which is used to slow down the convergence speed.

Even though DE algorithm show its robustness and efficiency in solving most optimization problems, pre-mature convergence may happens during the optimization process especially in solving the multi-modal function. DE probably falls into a traps of the optima and waste a large number of function evaluations to search such a narrow area around the local optima. The mutation strategy of DE presented in [8] is an example of avoiding the DE fall in the local optima with the help of an offset from two random selected individuals However it still needs a lot of FEs for DE to get rid of traps. The motivation of the RS for population vectors is to help the DE algorithm detect whether the population has fallen into the local optima or not and help population recover.

Every β interactions, RS for population vectors will evaluate whether the best function value has been improved significantly. Equation (9) provides a measure to find out this situation.

$$\frac{f(X_{best}^{t})-f(X_{best}^{last})}{f(X_{best}^{t})}<\varepsilon \tag{9}$$

Where the X_{best}^{t} is the current best individual and the X_{best}^{last} is the best individual in last β interactions. In the (9) , ε is a constant value in the range between 0 to 1.

Once the condition presented in (9) is achieved, the population will be replaced by a new population, which is generated by using the information provided by the current best individual X_{best}^{t} , and will be distributed around the current best individual since the RS for population uses a Gauss random number to set up the offset. The replacement equation (10) is the replacement strategy for every individual and every dimension.

$$X_{i,j}^{t+1}=X_{best}^{t}+\frac{x_{\max}-x_{\min}}{\delta}\cdot Gauss(0,1) \tag{10}$$

Where $X_{i,j}^{t+1}$ is i-th individual j-th dimension parameter in the new population, $Gauss(0,1)$ is a random number with expectation 0 and standard deviation and δ is a constant for locating the new population around the current best individual but not at the same position of it.

The strategy present above is based on the suppose that a high quality individual provide a high quality information to regenerate a population so that it can be future exploration and exploitation. However, if the current best individual is already in the traps, the performance of the RS is weak. To overcome such a situation, RSDE offer another type of the RS for population vectors. If the RS for population vectors achieves γ times continuously, which can be considered that the population has fallen into a unrecoverable trap, the population will be regenerated without using any historical information.

The second replacement strategy is aim to slow down the convergence speed and relocate the population around the current best individual for explore the area near it. If the population has not been improve significantly even though the RS for population vector has been run, the population will be regenerated using uniform random offset like the DE initial process and it will find another optima quickly because DE is well perform finding the solution of the objective function problem.

IV. Experimental Results

The RSDE algorithm was coded in Matlab 2012a and run on an Intel(R) Core(TM) i5-2450 CPU @ 2.50Hz with 4 GB TAM memory, under Windows 8.1 pro, 64 bit OS. Every independent run's initialization is based on the random parameter which is related to the time.

A. Experimental setting

IEEE-CEC 2014 single objective real-parameter optimization benchmark is used to test the performance of the RSDE. All details of these problems will be treated as black-box problem for the competition.

The experiment settings, which the RSDE test should follow, are carried out under the following details:

- The dimension of test optimization problem (minimization problems) has 4 types (10,30,50 and 100). All benchmark function with all dimension are tested in this paper.
- Maximum number of function evaluations : 1.0E+04*D
- The number of repetition optimization: 51 runs.
- Uniform random initialization within the search space. Random seed is based on time. As the RSDE is coded in Matlab, the seed is set as rand('state', sum(100*clock)).
- The error value is used to measure the RSDE performance. And the expression of error value is as follow

  $$errorvalue = f(x)-f(x^{*})$$

  Where $f(x)$ is the function value of the so far best individual and the $f(x^{*})$ is the minimization of the objective function. All error value is considered as 0 while it is less than 1.0E-08.
- Once the FE reach the maximum function evaluation or the so far best objective function error value is less than 1.0E-08, the RSDE algorithm will be terminated.

Regarding the RSDE parameter, mutation (MR) is taken as 0.65 and the crossover rates is taken as 0.9. The population was 50. The special parameter $\alpha,\beta,\gamma,\delta,\varepsilon$ are taken as 5*D (Dimension of the problem), 10*D , 3, 10 and 1e-4.

The performance evaluation of RSDE is conducted as the requirement paper provided by the CEC 2014 Special Session and Competition. Each function has been run 51 times and sort the Max FEs result from the smallest (best) to the largest (worst) and find out the best, worst, mean, median and standard variance values of the function error values, $f(x)-f(x^{*})$, for 51 runs. The results are present in Table 1 to 4. The convergence graphs for every function are illustrated in Figure 1 to 20. The complexity of the algorithm is introduced in the Table 5.

B. Result data analysis

Based on the data from Table 1 to Table 4, we can denote the performance of RSDE algorithm. The analysis for the result is as follows:

1) 10D benchmark function

10D dimension of CEC2014 single objective real-parameter numerical optimization is required for the initial submission as well as the 30D dimension problem. In table 1, we can denote that the RSDE successfully solve function 1, 2, 3, 4, 5, 6, 8 (where function 1, 2, 3, Unimodal function, is solved perfectly). And function 7, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 27 is solved under the 1. Obviously, RSDE has great performance in low dimension function problem, especially in the unimodal function.

2) 30D benchmark function

The problem will become increasingly different to be solved as the dimension is increased. The result of 30D is obviously worse than the 10D. However, function 2 and function 7 can still be solved and 5 functions (3, 4, 12, 13, 14) is less than 10^{-2}. The performance is acceptable.

3) 50D and 100D benchmark function

In the problem 50D and 100D, RSDE can not solve any of them. However, the result show that the RSDE has great influence on the population which will lead them to find out a better position in the function search space, a better vector parameter.

Table1: Best, worst, median, mean error value and Std value for RSDE of D = 10

Func.	Best	Worst	Median	Mean	Std
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	0.00E+00	3.48E+01	0.00E+00	2.81E+00	8.24E+00
5	0.00E+00	2.00E+01	2.00E+01	1.92E+01	3.92E+00
6	0.00E+00	9.03E-01	3.25E-06	5.29E-02	2.13E-01
7	1.04E-07	1.70E-01	2.47E-02	3.55E-02	3.12E-02
8	0.00E+00	3.42E+00	2.24E-04	6.61E-01	9.31E-01
9	1.99E+00	1.69E+01	2.99E+00	8.52E+00	3.71E+00
10	3.54E+00	2.53E+02	1.55E+02	6.84E+01	6.65E+01
11	1.86E+01	9.05E+02	4.14E+02	2.91E+02	1.93E+02
12	3.06E-02	5.76E-01	1.51E-01	2.21E-01	1.37E-01
13	6.16E-02	1.81E-01	7.36E-02	1.28E-01	3.18E-02
14	2.26E-02	2.19E-01	1.39E-01	1.36E-01	4.36E-02
15	3.64E-01	2.29E+00	7.05E-01	9.83E-01	3.70E-01
16	1.08E+00	3.16E+00	2.41E+00	2.23E+00	4.32E-01
17	2.09E-01	1.79E+02	1.59E+02	4.77E+01	5.52E+01
18	5.33E-02	5.49E+00	1.61E+00	2.00E+00	1.10E+00
19	1.24E-01	1.77E+00	5.40E-01	1.03E+00	3.55E-01
20	1.91E-03	2.81E+00	5.59E-01	7.21E-01	6.22E-01
21	5.69E-02	2.32E+01	6.60E-01	1.21E+00	3.33E+00
22	5.83E-02	2.14E+01	1.07E+00	1.17E+01	9.74E+00
23	3.29E+02	3.29E+02	3.29E+02	3.29E+02	2.78E-13
24	1.09E+02	1.34E+02	1.21E+02	1.19E+02	6.59E+00
25	1.09E+02	2.01E+02	1.45E+02	1.30E+02	1.93E+01
26	1.00E+02	1.00E+02	1.00E+02	1.00E+02	3.65E-02
27	7.85E-01	3.46E+02	3.00E+02	9.12E+01	1.40E+02
28	3.57E+02	6.01E+02	3.69E+02	3.87E+02	4.88E+01
29	1.32E+02	2.25E+02	2.23E+02	2.13E+02	2.59E+01
30	4.55E+02	1.15E+03	4.62E+02	5.05E+02	1.06E+02

Table2: Best, worst, median, mean error value and Std value for RSDE of D = 30

Func.	Best	Worst	Median	Mean	Std
1	4.60E+00	6.06E+03	1.01E+03	1.50E+03	1.70E+03
2	0.00E+00	3.34E-08	0.00E+00	0.00E+00	5.99E-09
3	3.89E-07	5.54E-01	1.13E-01	4.74E-02	1.16E-01
4	3.19E-07	7.35E+01	1.18E-02	3.05E+00	1.34E+01
5	2.01E+01	2.06E+01	2.02E+01	2.03E+01	9.88E-02
6	1.38E+00	9.83E+00	6.67E+00	5.16E+00	2.01E+00
7	0.00E+00	8.75E-03	9.50E-04	8.46E-04	1.59E-03
8	7.28E+00	3.93E+01	1.62E+01	2.04E+01	7.04E+00
9	1.89E+01	1.05E+02	6.27E+01	5.80E+01	1.65E+01
10	2.09E+01	1.07E+03	3.89E+02	3.29E+02	2.47E+02
11	1.01E+03	4.11E+03	2.73E+03	2.74E+03	6.44E+02
12	1.46E-01	8.84E-01	3.92E-01	4.44E-01	1.66E-01
13	1.77E-01	4.06E-01	3.00E-01	3.05E-01	5.50E-02
14	1.49E-01	3.07E-01	2.33E-01	2.36E-01	3.37E-02
15	2.04E+00	1.36E+01	3.88E+00	5.92E+00	2.59E+00
16	8.36E+00	1.21E+01	1.05E+01	1.06E+01	7.70E-01
17	4.25E+02	2.15E+03	1.22E+03	1.24E+03	3.79E+02
18	1.79E+01	1.89E+02	9.72E+01	9.54E+01	4.34E+01
19	3.05E+00	1.02E+01	5.11E+00	5.65E+00	1.46E+00
20	1.05E+01	1.65E+02	5.32E+01	3.73E+01	2.55E+01
21	9.07E+01	1.26E+03	5.54E+02	4.71E+02	2.34E+02
22	2.34E+01	4.50E+02	3.54E+01	1.91E+02	1.19E+02
23	3.15E+02	3.15E+02	3.15E+02	3.15E+02	1.62E-06
24	2.18E+02	2.27E+02	2.25E+02	2.24E+02	1.65E+00
25	2.03E+02	2.03E+02	2.03E+02	2.03E+02	1.17E-01
26	1.00E+02	1.00E+02	1.00E+02	1.00E+02	4.14E-02
27	3.47E+02	7.26E+02	5.33E+02	4.69E+02	9.46E+01
28	7.20E+02	1.30E+03	1.18E+03	9.05E+02	1.21E+02
29	7.21E+02	1.31E+07	7.94E+02	6.52E+05	2.66E+06
30	6.31E+02	4.46E+03	8.91E+02	1.70E+03	8.67E+02

Table3: Best, worst, median, mean error value and Std value for RSDE of D = 50

Func.	Best	Worst	Median	Mean	Std
1	2.80E+03	5.99E+04	5.99E+04	2.25E+04	1.21E+04
2	3.66E-03	2.51E+04	2.79E+02	3.58E+03	6.56E+03
3	2.03E-03	8.79E+00	7.09E-02	4.10E-01	1.38E+00
4	5.24E-02	1.56E+02	7.27E+01	6.41E+01	3.62E+01
5	2.02E+01	2.07E+01	2.05E+01	2.05E+01	9.61E-02
6	9.32E+00	2.98E+01	2.31E+01	1.72E+01	4.33E+00
7	2.95E-08	1.97E-02	4.36E-08	2.40E-03	4.40E-03
8	2.94E+01	7.90E+01	4.68E+01	5.04E+01	1.24E+01
9	8.26E+01	2.24E+02	1.12E+02	1.42E+02	3.51E+01
10	3.90E+02	4.30E+03	1.05E+03	1.52E+03	9.46E+02
11	3.12E+03	8.34E+03	6.85E+03	6.15E+03	1.05E+03
12	1.44E-01	1.11E+00	5.58E-01	5.38E-01	2.02E-01
13	3.07E-01	5.31E-01	4.21E-01	4.23E-01	6.05E-02
14	2.27E-01	3.16E-01	3.01E-01	2.78E-01	2.25E-02
15	4.22E+00	3.09E+01	8.32E+00	9.96E+00	6.53E+00
16	1.77E+01	2.16E+01	2.07E+01	1.93E+01	8.42E-01
17	1.38E+03	8.88E+03	6.11E+03	4.10E+03	1.68E+03
18	1.10E+02	1.38E+03	3.56E+02	3.40E+02	2.38E+02
19	1.05E+01	1.87E+01	1.49E+01	1.46E+01	2.03E+00
20	3.84E+01	3.14E+02	6.31E+01	1.60E+02	7.30E+01
21	5.55E+02	5.36E+03	1.32E+03	1.58E+03	6.55E+02
22	4.18E+01	1.31E+03	9.74E+02	4.61E+02	2.34E+02
23	3.44E+02	3.44E+02	3.44E+02	3.44E+02	1.19E-05
24	2.71E+02	2.82E+02	2.74E+02	2.76E+02	2.38E+00
25	2.05E+02	2.09E+02	2.07E+02	2.06E+02	7.81E-01
26	1.00E+02	3.60E+02	1.00E+02	1.12E+02	4.97E+01
27	6.07E+02	1.08E+03	8.74E+02	8.04E+02	1.00E+02
28	1.15E+03	2.52E+03	2.19E+03	1.61E+03	3.90E+02
29	8.34E+02	6.59E+07	1.77E+03	5.28E+06	1.64E+07
30	8.57E+03	1.97E+04	1.01E+04	1.12E+04	1.75E+03

Table4: Best, worst, median, mean error value and Std value for RSDE of D = 100

Func.	Best	Worst	Median	Mean	Std
1	2.72E+05	1.92E+06	9.32E+05	8.33E+05	2.89E+05
2	1.05E-02	4.26E+04	7.34E+01	7.39E+03	9.84E+03
3	2.64E-02	1.50E+01	2.56E-01	9.77E-01	2.21E+00
4	9.44E+01	2.97E+02	2.27E+02	1.86E+02	4.06E+01
5	2.06E+01	2.09E+01	2.09E+01	2.08E+01	7.85E-02
6	4.45E+01	8.71E+01	6.06E+01	6.02E+01	7.48E+00
7	2.15E-05	1.65E-02	1.41E-04	1.27E-03	2.71E-03
8	1.24E+02	2.55E+02	1.53E+02	1.94E+02	3.02E+01
9	1.90E+02	4.51E+02	3.44E+02	3.20E+02	5.41E+01
10	4.18E+03	1.60E+04	8.72E+03	9.31E+03	2.67E+03
11	1.25E+04	1.96E+04	1.64E+04	1.55E+04	1.54E+03
12	4.18E-01	1.32E+00	7.72E-01	7.42E-01	1.97E-01
13	4.17E-01	6.23E-01	5.04E-01	5.44E-01	4.09E-02
14	1.70E-01	2.34E-01	2.18E-01	2.09E-01	1.23E-02
15	1.99E+01	9.76E+01	3.44E+01	5.24E+01	1.82E+01
16	3.98E+01	4.48E+01	4.17E+01	4.24E+01	1.21E+00
17	3.07E+04	2.85E+05	1.24E+05	9.86E+04	4.60E+04
18	4.21E+02	6.04E+03	7.44E+02	1.26E+03	1.08E+03
19	3.25E+01	1.31E+02	1.02E+02	8.16E+01	2.58E+01
20	2.67E+02	1.07E+03	5.22E+02	5.50E+02	1.76E+02
21	7.87E+03	9.43E+04	7.87E+03	3.49E+04	1.90E+04
22	6.35E+02	2.54E+03	1.92E+03	1.51E+03	4.63E+02
23	3.48E+02	3.48E+02	3.48E+02	3.48E+02	2.12E-03
24	3.92E+02	4.16E+02	4.15E+02	4.06E+02	5.67E+00
25	2.28E+02	2.65E+02	2.44E+02	2.42E+02	7.50E+00
26	1.01E+02	4.90E+02	2.00E+02	1.98E+02	4.97E+01
27	1.61E+03	2.56E+03	2.13E+03	2.01E+03	1.65E+02
28	2.82E+03	5.69E+03	4.87E+03	4.11E+03	7.20E+02
29	1.23E+03	2.38E+08	2.00E+03	8.29E+07	8.20E+07
30	8.09E+03	1.94E+04	1.94E+04	1.31E+04	2.31E+03

Figure 1. Convergence figure of RSDE for D=10

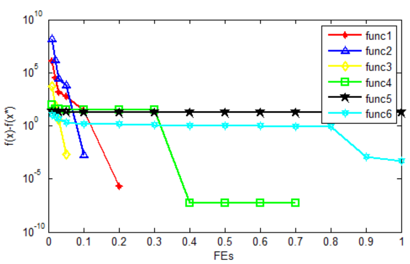


Figure 2. Convergence figure of RSDE for D=10

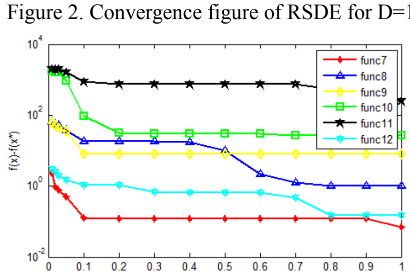


Figure 3. Convergence figureof RSDE for D=10

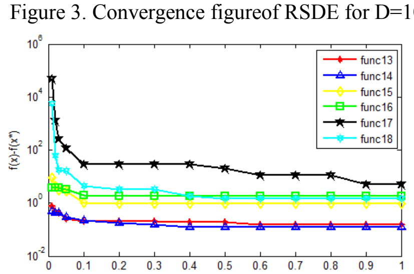


Figure 4. Convergence figureof RSDE for D=10

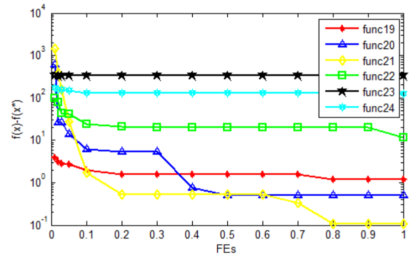


Figure 5.Convergence figureof RSDE for D=10

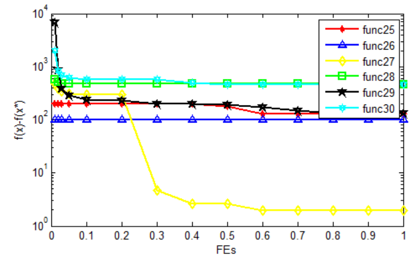


Figure 6. Convergence figureof RSDE for D=30

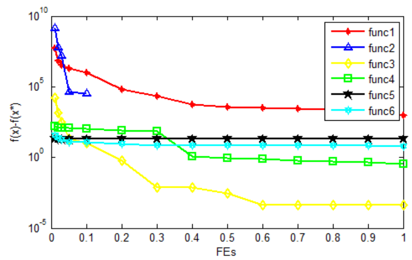


Figure 7. Convergence figureof RSDE for D=30

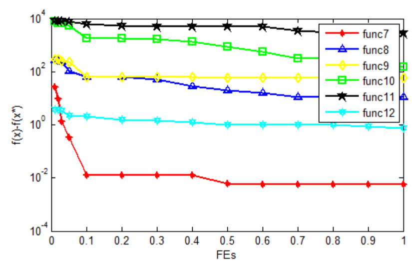


Figure 8. Convergence figureof RSDE for D=30

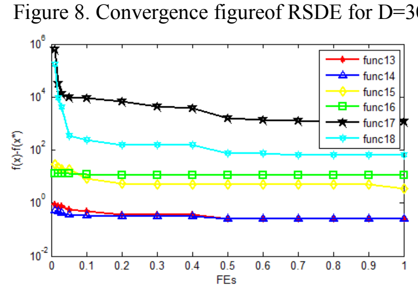


Figure 9. Convergence figureof RSDE for D=30

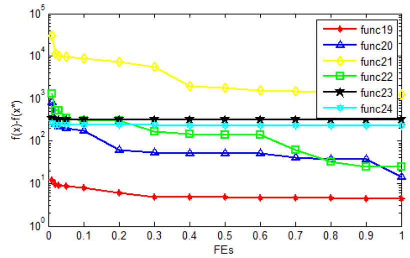


Figure 14. Convergence figureof RSDE for D=50

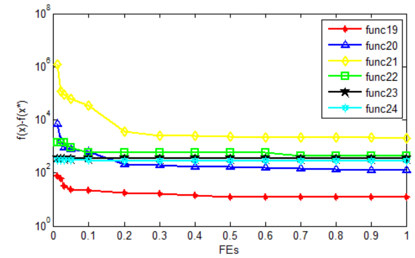


Figure 10. Convergence figureof RSDE for D=30

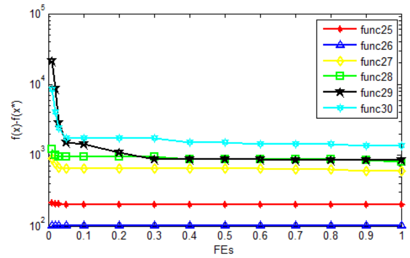


Figure 15. Convergence figureof RSDE for D=50

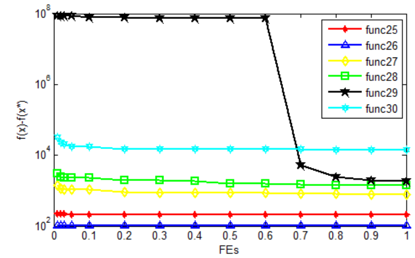


Figure 11. Convergence figureof RSDE for D=50

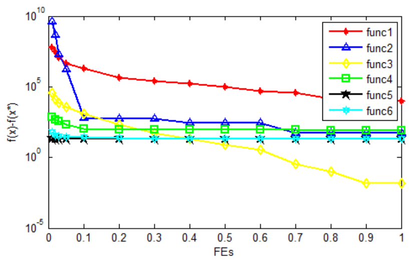


Figure 16. Convergence figureof RSDE for D=100

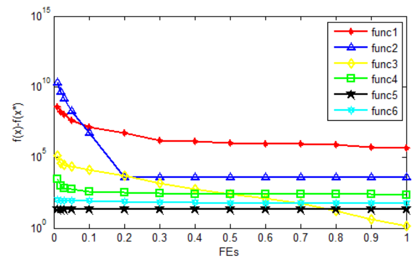


Figure 12. Convergence figureof RSDE for D=50

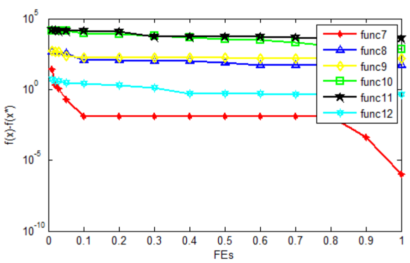


Figure 17. Convergence figureof RSDE for D=100

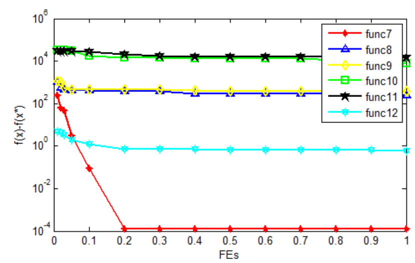


Figure 13. Convergence figureof RSDE for D=50

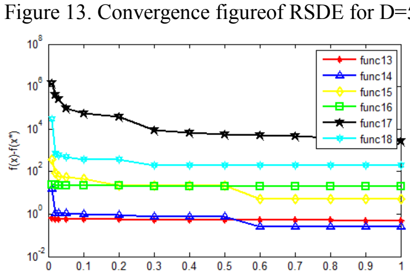


Figure 18. Convergence figureof RSDE for D=100

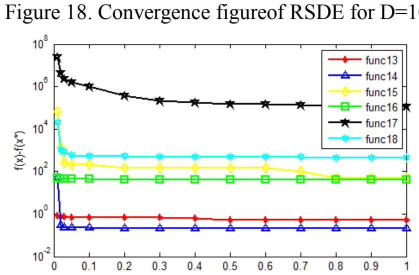


Figure 19. Convergence figureof RSDE for D=100

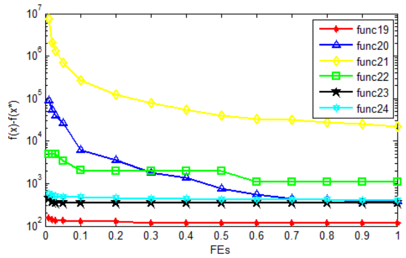


Figure 20. Convergence figureof RSDE for D=100

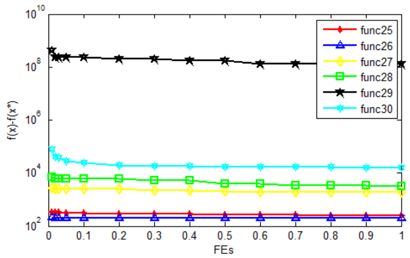


Table 5: the complexity of the RSDE algorithm

	10D	30D	50D
T0	0.1477	0.1477	0.1477
T1	3.9436	4.4075	4.5525
T2	7.6931	9.1191	9.3439
(T2 - T1) / T0	25.3880	31.9023	32.4426

C. Analysis of the convergence figure

For each figure, the X axis is denoted as the FEs and the Y axis, using the exponent presentation, respect the error value of the responding function. Every 0.1 maximum of Function evaluation, the program will mark down the error value in the algorithm process.

In the convergence figure of 10D, the RSDE solve the Uni-modal function before the maximum function evaluation meet and the convergence is rapidly. Because the convergence figure provided in last subsection is based on the so far best individual, it won't reflect the details of the RSDE evolution process. The figure 21 and figure 22 is presented to show the detail and the feature of RSDE evolutionary process. It's obvious the population has been regenerated several times so that the population reset in a better position to find out a better parameter vector solution.

Figure 21. evolution process of RSDE for D=10 in function 8

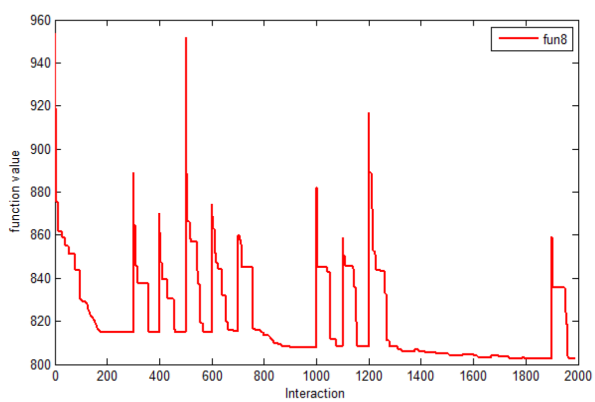
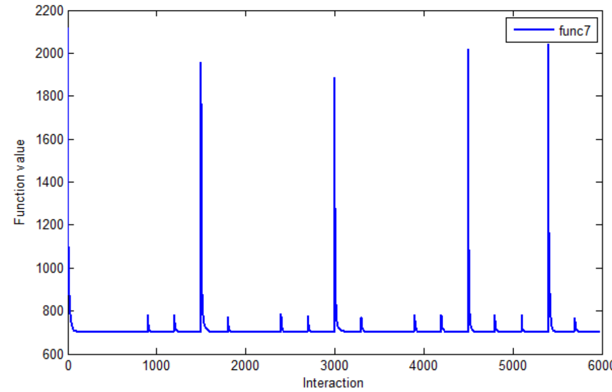


Figure 21. evolution process of RSDE for D=30 in function 7



V. CONCLUSION

In this paper, we proposed an improved DE algorithm with the replacement strategy denoted as RSDE. The individual replacement strategy and the population replacement strategy improve the DE exploitation performance and exploration performance respectively. A result report is presented for the CEC 2014 Special Session and Competition on Real-parameter Numerical Optimization. The result shows that RSDE performs well in the Uni-modal functions and most Multi-modal functions.

VI. ACKNOWLEDGE

This work was supported by National Natural Science Foundation of China (61370102, 61170193), Guangdong Natural Science Foundation (S2012010010613), the Fundamental Research Funds for the Central Uni- versities, SCUT (2012ZZ0087) and The Pearl River Science & Technology Star Project (2012J2200007).

REFERENCES

[1] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces.Journal of global optimization, 11(4), 341-359.
[2] Chiou, J. P., & Wang, F. S. (1998, May). A hybrid method of differential evolution with application to optimal control problems of a bioprocess system.In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on (pp. 627-632). IEEE.
[3] Ilonen, J., Kamarainen, J. K., &Lampinen, J. (2003). Differential evolution training algorithm for feed-forward neural networks. Neural Processing Letters, 17(1), 93-105.
[4]Rocca, P., Oliveri, G., & Massa, A. (2011). Differential evolution as applied to electromagnetics. Antennas and Propagation Magazine, IEEE, 53(1), 38-49.
[5]Price K, Storn R M, Lampinen J A. Differential evolution: a practical approach to global optimization[M]. Springer, 2006.
[6]Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. Evolutionary Computation, IEEE Transactions on, 2009, 13(2): 398-417.
[7] Zhang J, Sanderson A C. JADE: adaptive differential evolution with optional external archive[J]. Evolutionary Computation, IEEE Transactions on, 2009, 13(5): 945-958.
[8] Mezura-Montes E, Velázquez-Reyes J, Coello Coello C A. Modified differential evolution for constrained optimization[C]//Evolutionary Computation, 2006. CEC 2006. IEEE Congress on. IEEE, 2006: 25-32.
[9] Gämperle R, Müller S D, Koumoutsakos P. A parameter study for differential evolution[J]. Advances in intelligent systems, fuzzy systems, evolutionary computation, 2002, 10: 293-298.