Experimental Study on Differential Evolution Strategies

Youyun Ao[1], Hongqin Chi[2]
[1]School of Computer and Information,
Anqing Teachers College, Anqing, Anhui, 246011, China
[2]Mathematics and Science College,
Shanghai Normal University, Shanghai, 200234, China
youyun.ao@gmail.com, chihq@shnu.edu.cn

Abstract

Differential evolution (DE) algorithm has been proven to be a simple and efficient evolutionary algorithm for global optimization over continuous spaces, which is widely used in both benchmark test functions and real-world applications. Like genetic algorithms, differential evolution algorithm uses three typical operators to search the solution space: crossover, mutation and selection. Among these operators, mutation plays a key role in the performance of differential evolution algorithm and there are several mutation variants often used, which constitute several corresponding differential evolution strategies. By means of experiments, this paper investigates the relative performance of different differential evolution algorithms for global optimization under different differential evolution strategies respectively. In simulation studies, De Jong's test functions have been employed, and some conclusions are drawn.

1. Introduction

Differential Evolution (DE) algorithm is a simple yet effective evolutionary algorithm like genetic algorithms for global optimization over continuous spaces [1], [2], and has recently been shown to produce superior results in both widely used benchmark functions [3] and real-world applications [4], [5]. In the differential evolution algorithm, there are conventionally three operators (i.e., mutation, crossover and selection) and three parameters, i.e., the population size NP, the scale factor F and the crossover probability CR. Mutation plays a key role in the performance of the differential evolution algorithm and there are several variants of mutation. The selection of mutation variants and parameters is the most important issue of the differential evolution algorithm research. Several mutation variants and theirs parameters constitute several corresponding differential evolution strategies [6], [7]. Experimental parameter studies and empirical parameter settings of the differential evolution algorithm have been carried out [8]. This paper investigates the relative contribution of several differential evolution strategies based on several variants of mutation to differential evolution algorithms for global optimization. The aim is to help users to select one with the best performance from candidate mutations under certain environment.

Without loss of generality, a minimization problem is formulated as follows [9]: minimize $f(\vec{x}) = f(x_1, x_2, ..., x_D)$, subject to $g_j(\vec{x}) \le 0, j = 1,2,...,q$ and $h_j(\vec{x}) = 0, j = q+1,...,m$, where $l_j \le x_j \le u_j, j \in \{1,2,...,D\}$, the minimization problem is to find $\vec{x}^*$ such that $f(\vec{x}^*) \le f(\vec{x})$.

The rest of this paper is organized as follows. Firstly, we review the basic differential evolution algorithm in Section 2. And then, we describe the different differential evolution strategies in Section 3. Section 4 investigates the relative performance of different differential evolution strategies by means of experiments. Finally, conclusions are drawn in Section 5.

2. Differential evolution algorithm

Let $\vec{x}^i_{G=t} = (x^i_{1,G=t}, ..., x^i_{D,G=t})$ denotes an individual i, a population $P_{G=t}$ at generation $G = t$ is a set of NP individuals ($NP > 4$). According to the description by Storn and Price [1], [6], [7], [10], the classical differential evolution algorithm can be outlined in the following.

2.1. Initial population at random

The initial population $P_{G=0} = \{\vec{x}^1_{G=0}, ..., \vec{x}^{NP}_{G=0}\}$ is initialized as $\forall i \le NP, \forall j \le D : \vec{x}^i_{j,G=0} = l_j + \mathrm{r}_j \times (u_j - l_j)$, where NP is the population size, D is the individual's

978-0-7695-3571-5/09 $25.00 © 2009 IEEE
DOI 10.1109/GCIS.2009.31

dimension, $r_j \in [0,1]$ is a random number, and $\vec{x}^{i}_{j,G=0}$ is one variable j in an individual i at initial generation $G=0$, which is initialized within its boundaries $[l_i, u_i]$.

2.2. Mutation operation

For each individual $\vec{x}^{i}_{G=t} = (x^{i}_{1,G=t},\dots,x^{i}_{D,G=t})$ in the population $P_{G=t}$, a new individual is generated by the following formula:

$$\vec{y}^{i}_{G=t} = \vec{x}^{r_3}_{G=t} + F \times (\vec{x}^{r_1}_{G=t} - \vec{x}^{r_2}_{G=t}),\ i = 1,2,\dots,NP$$

with $r_1, r_2, r_3 \in [1, NP]$, three mutually different integers and they are also different from the running index i.The scale factor $F > 0$ is a real constant and is often set to 0.5.

2.3. Crossover operation

The new individual is generated by recombining $\vec{x}^{i}_{G=t} = (x^{i}_{1,G=t},\dots,x^{i}_{D,G=t})$ and $\vec{y}^{i}_{G=t} = (y^{i}_{1,G=t},\dots,y^{i}_{D,G=t})$ according to the following formula:

$$\vec{z}^{i}_{j,G=t} = \begin{cases} \vec{y}^{i}_{j,G=t},\ if\ (random[0,1) \le CR \mid j = j_{rand}) \\ \vec{x}^{i}_{j,G=t},\ otherwise \end{cases}$$

where $random[0,1)$ stands for the uniform random number at intervals $[0,1)$, and j_{rand} is a randomly chosen index to ensure that at least one of the variables should be changed and $\vec{z}^{i}_{G=t}$ does not duplicate $\vec{x}^{i}_{G=t}$. And the crossover probability $CR \in (0,1)$ is often set to 0.9.

2.4. Selection operation

If the new individual is better than the original one then the new individual is to be an offspring in the next generation $G = t+1$ else the new individual is discarded and the original one is retained in the next generation.

$$\vec{x}^{i}_{G=t+1} = \begin{cases} \vec{z}^{i}_{G=t},\ if\ (\vec{x}^{i}_{G=t}) > f(\vec{z}^{i}_{G=t}) \\ \vec{x}^{i}_{G=t},\ otherwise \end{cases}$$

where $f(\cdot)$ is the fitness function and $\vec{x}^{i}_{G=t+1}$ is the offspring of $\vec{x}^{i}_{G=t}$ in the next generation.

2.5. Algorithm framework

Input: Parameters of the algorithm.
Step 1. Initialize the initial population $P_{G=0}$.
Step 2. Evaluate $P_{G=0}$ and let the generation counter $t = 0$.
Step 3. While (*the stopping criterion is not satisfied*) do
{For each individual $\vec{x}^{i}_{G=t}$, $i = 1,2,\dots,NP$, its offspring $\vec{x}^{i}_{G=t+1}$ is generated by mutation, crossover and selection operations.
Evaluate $P_{G=t+1}$ and let $t = t+1$.
} End While
Output: The best solution $\vec{x}^{*}_{G=t}$ from $P_{G=t}$.

3. Differential evolution strategies

In the differential evolution algorithm, there are several variants of mutation, which constitute the several corresponding differential evolution strategies. The most useful differential evolution strategies are described in the following [7], [9], [11], [12], [16].

3.1. Strategy DE/rand/1

For each individual $\vec{x}^{i}_{G=t}$ from population $P_{G=t}$ of size NP, $i = 1,2,\dots,NP$, a perturbed individual $\vec{y}^{i}_{G=t}$ is generated by the following formula:

$$\vec{y}^{i}_{G=t} = \vec{x}^{r_3}_{G=t} + F \times (\vec{x}^{r_1}_{G=t} - \vec{x}^{r_2}_{G=t})$$

with $r_1, r_2, r_3 \in [1, NP]$, three mutually different integers. The random integers r_1, r_2 and r_3 are also chosen to be different from the running index i. The scale factor $F \in [0,2]$ is a real constant, which is used to control the amplification of the differential variation $(\vec{x}^{r_1}_{G=t} - \vec{x}^{r_2}_{G=t})$.

3.2. Strategy DE/best/1

Strategy DE/best/1 works the same way as strategy DE/rand/1 except that it generates the individual $\vec{y}^{i}_{G=t}$ according to the following formula:

$$\vec{y}^{i}_{G=t} = \vec{x}^{*}_{G=t} + F \times (\vec{x}^{r_1}_{G=t} - \vec{x}^{r_2}_{G=t})$$

the individual to be perturbed is the best individual $\vec{x}^{*}_{G=t}$ of the current generation $G = t$.

3.3. Strategy DE/best/2

Strategy DE/best/2 uses two difference individuals as a perturbation according to the following formula:

$$\vec{y}^{i}_{G=t} = \vec{x}^{*}_{G=t} + F \times (\vec{x}^{r_1}_{G=t} - \vec{x}^{r_2}_{G=t}) + F \times (\vec{x}^{r_3}_{G=t} - \vec{x}^{r_4}_{G=t})$$

with $r_1, r_2, r_3, r_4 \in [1, NP]$, four mutually different integers. The random integers r_1, r_2, r_3 and r_4 are also chosen to be different from the running index i.

3.4. Strategy DE/current to best/2

Strategy DE/current to best/2 also uses two difference individuals, but one between the best individual $\vec{x}^{*}_{G=t}$ of the current generation $G=t$ and the current individual $\vec{x}^{i}_{G=t}$ as follows:

$$\vec{y}^{i}_{G=t}=\vec{x}^{i}_{G=t}+F\times(\vec{x}^{*}_{G=t}-\vec{x}^{i}_{G=t})+F\times(\vec{x}^{r_1}_{G=t}-\vec{x}^{r_2}_{G=t})$$

The individual to be perturbed is the current individual $\vec{x}^{i}_{G=t}$ of the current generation $G=t$.

3.5. Strategy DE/rand/2

Strategy DE/rand/2 uses two difference individuals according to the following formula:

$$\vec{y}^{i}_{G=t}=\vec{x}^{r_5}_{G=t}+F\times(\vec{x}^{r_1}_{G=t}-\vec{x}^{r_2}_{G=t})+F\times(\vec{x}^{r_3}_{G=t}-\vec{x}^{r_4}_{G=t})$$

with $r_1,r_2,r_3,r_4,r_5\in[1,NP]$, five mutually different integers. The random integers r_1, r_2, r_3, r_4 and r_5 are also chosen to be different from the running index i.

4. Experimental study

4.1. Test functions used in simulation studies

In order to investigate the relative contribution of strategies DE/rand/1, DE/best/1, DE/best/2, DE/current to best/2 and DE/rand/2 to differential evolution algorithms, we use five benchmark test functions (f_1-f_5), which have been firstly presented by De Jong [2], [13] to test the performance of genetic algorithms. Thereafter, these minimization functions have been extensively used by genetic algorithm and other algorithm researchers [14], [15]. The test environment includes functions which are convex (f_1), non-convex (f_2), discontinuous (f_3), stochastic (f_4), and multimodal (f_5).

1) First De Jong function f_1 (Sphere)

$$f_1(\vec{x})=\sum_{j=1}^{3}x_j^2,\ x_j\in[-5.12,5.12]$$

The first minimization function is smooth, unimodal and strongly convex. The global optimum is $f_1(0)=0$.

2) Second De Jong function f_2 (Rosenbock's Saddle)

$$f_2(\vec{x})=100(x_1^2-x_2)^2+(1-x_1)^2,\ x_j\in[-2.048,2.048]$$

The second minimization function is considered to be a difficult problem and has been repeatedly used to test the performance of many optimization algorithms. The global optimum is $f_2(1)=0$.

3) Third De Jong function f_3 (Step)

$$f_3(\vec{x})=30+\sum_{j=1}^{5}\lfloor x_j\rfloor,\ x_j\in[-5.12,5.12]$$

The third minimization function has many flat surfaces, which makes many optimization algorithms easily get stuck on one of the flat plateaus. The minimum is $f_3(-5-\varepsilon)=0$, where $\varepsilon\in[0,0.12]$.

4) Modified fourth De Jong function f_4 (Quartic)

$$f_4(\vec{x})=\sum_{j=1}^{30}(j\cdot x_j^4+\eta),\ x_j\in[-1.28,1.28]$$

The fourth minimization function is a problem in the presence of noise, which makes an optimization algorithm hardly get the same value on the same point. Here, η is a random variable generated by the uniform distribution in the range $[0,1]$, and the inclusion of η makes the function a difficult problem. The minimum is $f_4(\vec{0})\le 30\cdot E(\eta)=15$, where $E(\eta)$ is the expectation of η.

5) Fifth De Jong function f_5 (Shekel's Foxholes)

$$f_5(\vec{x})=\frac{1}{0.002+\psi(\vec{x})},\ x_j\in[-65.536,65.536]$$

where $\psi(\vec{x})=\sum_{i=0}^{24}\frac{1}{1+i+\sum_{j=1}^{2}(x_j-a_{ij})^6}$

$a_{i1}=\{-32,-16,0,16,32\}$, for $i=\{0,1,2,3,4\}$ and $a_{i1}=a_{i\bmod 5,1}$

$a_{i2}=\{-32,-16,0,16,32\}$, for $i=\{0,5,10,15,20\}$ and

$a_{i2}=a_{i+k,2},\ k=\{1,2,3,4\}$

The fifth minimization function has many local optima. Many optimization algorithms easily get stuck in the first peak they find. The global minimum is $f_5(-32,-32)\cong 0.998004$.

4.2. Experimental Results

In this study, we firstly test the relative convergence speed of five differential evolution strategies DE/rand/1, DE/best/1, DE/best/2, DE/current to best/2 and DE/rand/2 on the first De Jong function f_1 using the following same parameter values: the population size $NP=10*D$, the scale factor $F=0.5$ and the crossover probability $CR=0.9$ according to the empirical parameter settings [7], [11], where D is the dimension of test function in the parameter space. Each differential evolution algorithm based on each differential evolution strategy is run repeatedly 30 times until the global optimum is found in each single run, when the generation number is used to evaluate the relative convergence speed. The mean generation number of 30 independent runs is summarized in Table 1.

Table 1. Mean generation number of f_1 under the first group of parameter values

Strategy	Parameter Values & Generation Number			
	NP	***F***	***CR***	***Generations***
DE/rand/1	10*D	0.5	0.9	2172
DE/best/1	10*D	0.5	0.9	885
DE/best/2	10*D	0.5	0.9	1533
DE/current to best/2	10*D	0.5	0.9	1312
DE/rand/2	10*D	0.5	0.9	2900

Table 1 shows that the differential evolution algorithm based on strategy DE/rand/1 can converge to the global optimum at mean 2172 generations, DE/best/1, DE/best/2, DE/current to best/1 and DE/rand/2 at mean 885, 1533, 1312 and 2900 generations respectively. Therefore, the differential evolution algorithm based on strategy DE/best/1 has the fastest convergence speed on the first De Jong test function f_1, and then, DE/current to best/2, DE/best/2, DE/rand/1, and DE/rand/2 from fast to slow relatively.

In order to study the impact of chosen parameter values on the convergence speed, we change the parameter values as follows: the population size $NP = 10 * D$, the scale factor $F = 0.5$ and the crossover probability $CR = 0.2$. The mean results of 30 independent runs are given in Table 2.

Table 2. Mean generation number of f_1 under the second group of parameter values

Strategy	Parameter Values & Generation Number			
	NP	***F***	***CR***	***Generations***
DE/rand/1	10*D	0.5	0.2	2330
DE/best/1	10*D	0.5	0.2	1385
DE/best/2	10*D	0.5	0.2	2033
DE/current to best/2	10*D	0.5	0.2	2136
DE/rand/2	10*D	0.5	0.2	2861

Table 2 shows that the differential evolution algorithm based on strategy DE/rand/1 can obtain the global optimum at mean 2330 generations, DE/best/1, DE/best/2, DE/current to best/2, and DE/rand/2 at mean 1385, 2033, 2136, and 2861 generations respectively. Hence, the sort of the relative convergence speed from fast to slow is DE/best/1, DE/best/2, DE/current to best/2 and DE/rand/2. Compared with the experimental results given in Table 1, we can find that DE/best/2 and DE/current to best/2 are slightly different in terms of the sort of the relative convergence speed. And we can also find that crossover probability has a stronger impact on DE/best/1, DE/best/2, and DE/current to best/2 than DE/rand/1 and DE/rand/2. A bigger crossover probability is helpful to improving the convergence speed of strategy DE/best/1, DE/best/2 or DE/current to best/2 due to employing the best individual.

Thereafter, we use the population size $NP = 20 * D$ or the scale factor $F = 0.6$ respectively, and the mean results are presented in Table 3 and Table 4 respectively.

Table 3. Mean generation number of f_1 under the third group of parameter values

Strategy	Parameter Values & Generation Number			
	NP	***F***	***CR***	***Generations***
DE/rand/1	20*D	0.5	0.9	2191
DE/best/1	20*D	0.5	0.9	867
DE/best/2	20*D	0.5	0.9	1510
DE/current to best/2	20*D	0.5	0.9	1315
DE/rand/2	20*D	0.5	0.9	2924

Compared with the statistical results given in Table 1, the mean generation number provided with Table 3 is slightly different. It is shown that the population size $NP = 10 * D$ is feasible. Generally, if the population size is too small, differential evolution algorithm has a bigger probability of getting stuck in the local optimum.

Table 4. Mean generation number of f_1 under the fourth group of parameter values

Strategy	Parameter Values & Generation Number			
	NP	***F***	***CR***	***Generations***
DE/rand/1	10*D	0.6	0.9	2482
DE/best/1	10*D	0.6	0.9	1161
DE/best/2	10*D	0.6	0.9	2157
DE/current to best/2	10*D	0.6	0.9	1413
DE/rand/2	10*D	0.6	0.9	3614

From Table 1 and Table 4, we can find that once the scale factor is slightly changed, the generation number is greatly changed. It is shown that the scale factor F can work on the convergence speed largely. Hence, the scale factor F is suitably chosen to be very important for the performance of differential evolution algorithm.

For the second De Jong test function f_2 , we firstly attempt to adopt the first group of parameter values used in the process of testing the first function f_1 : the population size $NP = 10 * D$, the scale factor $F = 0.5$ and the crossover probability $CR = 0.9$. However, we find that the differential evolution algorithm based on strategy DE/rand/1 either obtains the global optimum very quickly or gets stuck in the local optimum. Therefore, we modify the parameter values, and the test results are presented in Table 5 and Table 6 respectively.

Table 5. Mean generation number of f_2 under the first group of parameter values

Strategy	Parameter Values & Generation Number			
	NP	*F*	*CR*	*Generations*
DE/rand/1	10**D*	0.5	0.2	584
DE/best/1	10**D*	0.5	0.2	288
DE/best/2	10**D*	0.5	0.2	480
DE/current to best/2	10**D*	0.5	0.2	586
DE/rand/2	10**D*	0.5	0.2	762

Table 6. Mean generation number of f_2 under the second group of parameter values

Strategy	Parameter Values & Generation Number			
	NP	*F*	*CR*	*Generations*
DE/rand/1	10**D*	0.6	0.2	662
DE/best/1	10**D*	0.6	0.2	356
DE/best/2	10**D*	0.6	0.2	615
DE/current to best/2	10**D*	0.6	0.2	588
DE/rand/2	10**D*	0.6	0.2	904

According to the experimental results given in Table 5 and Table 6, it is clear that the sort of convergence speed from fast to slow is still DE/best/1, DE/best/2 or DE/current to best/2, DE/rand/1 and DE/rand/2. Compared with the results provided with Table 5, the generation number under each differential evolution strategy in Table 6 is popularly improved after the scale factor is reset. Besides, we can also conclude that the convergence speed is highly related with the feature of problem according to the tables above.

For the De Jong test functions f_3 and f_4, the following parameter values are chosen: the population size $NP = 10 * D$, the scale factor $F = 0.5$ and the crossover probability $CR = 0.2$ and the maximum generation number is set to 3000.

The experimental results of test function f_3 over 30 runs show that each differential evolution algorithm based on each differential evolution strategy can find the global minimum 0 in each single run.

The experimental results of test function f_4 over 30 runs show that each differential evolution algorithm based on each differential evolution strategy can obtain the function value which is less than 15 and not single due to the presence of noise in each single run.

In the following, the fifth De Jong test function f_5 is employed to test the relative convergence rate of each differential evolution algorithm based on each differential evolution strategy. We firstly set the first group of parameter values: the population size $NP = 10 * D$, the scale factor $F = 0.5$, the crossover probability $CR = 0.2$ and the maximum generation number is set to 3000. The statistical results over 30 independent runs are given in Table 7.

Table 7. Experimental results of f_5 under the first group of parameter values

Strategy	Error Results	
	Error Ratio	*Error Values*
DE/rand/1	0.0333333333333333	1.99203163396214
DE/best/1	0.2	1.99203163396214 ± 0
DE/best/2	0.166666666666667	1.99203163396214 ± 0
DE/current to best/2	0.1	1.99203163396214 ± 0
DE/rand/2	0	—

According to the *Error Ratio* provided with Table 7, we can find that the differential evolution algorithm based on strategy DE/rand/1, DE/best/1, DE/best/2, or DE/current to best/2 respectively has some probability of getting stuck in the local optimum, when the objective function value found is 1.99203163396214. The sort of *Error Ratio* from big to small is DE/best/2, DE/best/2, DE/current to best/2, DE/rand/1, and DE/rand/2. Hence, the corresponding sort of convergence rate is DE/rand/2, DE/rand/1, DE/current to best/2, DE/best/2 and DE/best/1.

Here, *Error Ratio* is the ratio of the times no found the global optimum and the total runs 30. And *Error Values* are the best function values found, including the mean and variance. Additionally, the global minimum function value found by strategy DE/rand/1, DE/best/1, DE/best/2, DE/current to best/2 or DE/rand/2 respectively all is 0.998003932264462.

Further, we investigate the relative convergence rate under the second group of parameter values: the population size $NP = 10 * D$, the scale factor $F = 0.6$, the crossover probability $CR = 0.2$ and the maximum generation number is set to 3000. The statistical results are given in Table 8.

Table 8. Experimental results of f_5 under the second group of parameter values

Strategy	Error Results	
	Error Ratio	*Error Values*
DE/rand/1	0.0333333333333333	1.99203163396214 ± 0
DE/best/1	0.233333333333333	1.99203163396214 ± 0
DE/best/2	0.166666666666667	1.99203163396214 ± 0
DE/current to best/2	0.0333333333333333	1.99203163396214 ± 0
DE/rand/2	0.0333333333333333	1.99203163396214 ± 0

Table 8 shows that the sort of convergence rate from big to small is DE/rand/2, DE/rand/1 or DE/current to best/2, DE/best/2, and DE/best/1, which holds the same sort as that of Table 7.

Thereafter, we use the third group of parameter values as follows: the population size $NP = 10 * D$, the scale factor $F = 0.5$, the crossover probability $CR = 0.9$ and the maximum generation number is set to 2000.The results are given in Table 9.

Table 9. Experimental results of f_5 under the third group of parameter values

Strategy	Error Results	
	Error Ratio	***Error Values***
DE/rand/1	0.0333333333333333	1.99203163396214 ± 0
DE/best/1	0.566666666666667	1.99203163396214 ± 0
DE/best/2	0.2	1.99203163396214 ± 0
DE/current to best/2	0.233333333333333	1.99203163396214; 1.99203163396214; 1.04816525460948; 1.02072162994407; 1.99203163396214; 1.99203163396214; 1.99203163396214
DE/rand/2	0.0333333333333333	1.99203163396214 ± 0

According to the *Error Ratio* given in Table 9, it is clear that the sort of convergence rate from high to low is DE/rand/2 or DE/rand/1, DE/best/2, DE/current to best/2, and DE/best/1, which is similar to that of Table 7 and Table 8, except that the sort between DE/best/2 and DE/current to best/2 is slightly different.

5. Conclusions

In this paper, we use De Jong's five representative test functions to investigate the relative performance of five strategies DE/rand/1, DE/best/1, DE/best/2, DE/current to best/2, and DE/rand/2. The aim is to help users to select one with the best performance from candidate mutation variants under certain environment. The experimental results generally show that strategies DE/best/1, DE/best/2 and DE/current to best/2 can relatively improve the convergence speed, but also are ease of getting stuck in the local optimum due to the introduction of the best individual, and that strategies DE/rand/2 and DE/rand/1 can relatively enhance the convergence rate of finding the global optimum, but also have a relatively slow convergence speed. Additionally, it turns out that the performance of differential evolution algorithm is very sensitive to the choice of parameters and is related with the feature of problem.

References

[1] R. Storn and K. Price, "Differential evolution – A simple and efficient heruistic for global optimization over continuous spaces", *Journal of Global Optimization*, vol.11, pp.341-359, 1997.

[2] R. Storn and K. Price, "Differential evolution – A simple and efficent adaptive scheme for global optimization over continuous spaces", *Technical Report TR-95-012*, Berkeley: International Computer Science Institute, 1995.

[3] J. Vesterstrom, R. Thomsen, "A comparative study of differential evolution, paraticle swarm optimization, and evolutionary algorithms on numerical benchmark problems", *Evolutionary Computation*, vol.2, pp.1980-1987, 2004.

[4] R. Storn, "System design by constraint adaptation and differential evolution", *IEEE Transactions on Evolutionary Computation*, vol.2, pp.82–102, 1999.

[5] R. Thomsen, "Flexible ligand docking using differential evolution", *Proceedings of the 2003 Congress on Evolutionary Computation*, vol.4, pp.2354–2361, 2003.

[6] Z. Yang, K. Tang and X. Yao, "Differential evolution for high-dimensional function optimization", *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC'07)*, Singapore, 25-28 September 2007, pp.3523-3530.

[7] Z. Yang, K. Tang and X. Yao, "Self-adaptive differential evolution with neighborhood search", *Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC2008)*, Hongkong, China, June 2008, pp.1110-1116. IEEE Press.

[8] R. Gamperle, S. D. Muller, and P. Koumoutsakos, "A parameter study for differential evolution", *Proceedings WSEAS International Conference On Advances In Intelligent Systems, Fuzzysystems, Evolutionary Computation*, pp.293-298, 2002.

[9] J. Brest, V. Žumer, and M. S. Maučec, "Selft-adaptive differential evolution algorithm in constrained real-parameter optimization", *2006 IEEE Congress on Evolutionary Computation*, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, July16-21, 2006, pp.918-926.

[10] H. A. Abbass, R. Sarker, C. Newton, "PDE: A Pareto-frontier differential evolution approach for multiobjective optimization problems", *Proceedings of IEEE Congress on Evolutionary Computation*, IEEE, Piscataway, NJ, 2001, 2:971-978.

[11] R. Storn, "On the usage of differential evolution for function optimization", *1996 Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS 1996)*, pp.519-523. IEEE,1996.

[12] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization,* Springer-Verlag, ISBN:3-540-20950-6, 2005.

[13] K. A. De Jong, *An Analysis Of The Behavior Of A Class Of Genetic Adaptive Systems*, Phd thesis, University of Michigan, Dissertation Abstracts International 36(10), 5140B. (University Microfilms No. 76-9381), 1975.

[14] D. Karaboğa, S.Ŏ kdem, "A simple and global optimization algorithm for engineering problems: differential evolution algorithm", *Turk.J.Elec.Engin*, vol.12, no.1, 2004, pp.53-60.

[15] D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, M. N. Vrahatis, "Parallel differential evolution", *Proceedings of the 2004 congress on evolutionary computation(CEC2004)*, June 19-23, 2004, Portland OR, USA.

[16] A. Qin and P. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization", *Proceedings of the Congress on Evolutionary Computation 2005 (CEC'2005)*, vol.1, pp.630-636, Piscataway, New Jersey, September 2005. Edinburgh, UK, IEEE Service Center.