

An Improved Differential Evolution Algorithm with Novel Mutation Strategy

Xin Shen^{1,a,*}, Dexuan Zou^{1,b} and Xin Zhang^{1,c}

¹School of Electrical Engineering and Automation, Jiangsu Normal University, Xuzhou 221116, China;
^ashenxinpassion@163.com, ^bzoudexuan@163.com, ^cZhangXin_1994 @126.com

Keywords: differential evolution algorithm, fitness value, trapped solutions, dynamic adjustment.

Abstract: Aiming at the defects of differential evolution, such as premature convergence, low accuracy and other shortcomings, an improved differential evolution algorithm with novel mutation strategy(NMSIDE) is presented. The novel mutation strategy is used to avoid being trapped into local minima for NMSIDE. If the evolution of the individual stagnates, the individual will rely on the current best individual to move closer to the global best individual. The mutation rate varies dynamically within the range of values, and the crossover rate is dynamically varied based on the number of iterations. NMSIDE is tested on 11 standard functions and compared with the other state-of-the-art algorithms. The experimental results show that NMSIDE has higher convergence precision, faster convergence speed and better robustness.

1. Introduction

Differential evolution algorithm [1] was first proposed by Storn and Price, in 1995. It has strong global optimization ability and robustness, which is better than many other traditional optimization approaches on solving some benchmark functions. Thus, DE becomes a popular algorithm in the field of optimization algorithms. At present, DE has been widely used in constrained optimization [2], control [3], economic load dispatch of power system [4], and so on.

Like many other optimization algorithms [5,6], DE is easy to trap into local minima and generate low accuracy when the problem is especially high-dimensional or nonlinear time-varying, so many scholars put forward the corresponding the improvements. The improvements of scholars mainly focus on the following three aspects. Firstly, the mutation strategy is improved. Zhang et al [7] proposed adaptive differential evolution with optional external archive, which introduced a new mutation strategy in which an external archive was introduced to store poor solutions and improved the current best solution. It ensures the diversity of the population and improves the ability of the population to jump out of the local optima. Yi et al [8] proposed a hybrid mutation strategy with individuals with poor fitness values using DE/rand/1 strategy and individuals with good fitness values using DE/current-to-best/1 strategy to balance the exploration and exploitation of the algorithm. Secondly, parameter adjustment mechanism is researched. Islam et al [9] used parameter adaptation mechanism to change the parameter values of the next generation of the corresponding individuals by recording successive parameter for each generation. Zhou et al [10] introduced crossover rate sorting mechanism in which excellent individuals would be assigned a smaller crossover rate to keep good information. Thirdly, the good operator is embedded in the differential evolution algorithm. Zou et al [11] proposed a hybrid differential evolution-harmony search algorithm, which embedded the mutation and crossover operator of DE in the harmony search algorithm. It avoids the algorithm into local minima.

In order to further improve the convergence accuracy of the algorithm, overcome the premature convergence phenomenon and enhance the robustness of the algorithm. An improved differential evolution algorithm with novel mutation strategy(NMSIDE) is presented. The algorithm has the following three improvements. Firstly, a group of individuals are randomly selected. Then, the best

individual is selected and disturbance individuals are selected based on their fitness value. Secondly, the trapped solutions are updated to jump out of the local optima. Thirdly, the parameters are dynamically selected so as to satisfy different search phase of the algorithm, which solves the problem of setting control parameters of the algorithm. Through testing standard functions, results show that the NMSIDE is superior to some other DE algorithms.

2. Basic Differential Evolution Algorithm

Differential evolution algorithm is a parallel algorithm. It contains the following operations: initialization, mutation, crossover and selection. The detailed steps are as follows:

(1) Initialization

All candidate solutions $\{x_{i,0} = (x_{i1,0}, x_{i2,0}, \dots, x_{iD,0}) \mid i = 1, 2, \dots, Np\}$ are initialized randomly from a uniform distribution in the ranges $[L_{ij}, U_{ij}]$ ($j = 1, 2, \dots, D$). Where D is the dimension of the problem and Np is the population size. L_{ij} represents the lower bound of $x_{ij,0}$. U_{ij} represents the upper bound of $x_{ij,0}$.

(2) Mutation

DE generates mutated individuals by disturbing the target individual. The most common mutation strategy in DE is DE/rand/1/bin. In addition, DE/current-to-rand/1/bin, DE/best/1/bin, DE/current-to-best/1/bin, and so on are also very popular. Trial vector $v_{i,G}$ is calculated according to one of the following four equations:

DE/rand/1/bin:

$$v_{i,G} = x_{r_0,G} + F \cdot (x_{r_1,G} - x_{r_2,G}) \quad (1)$$

DE/current-to-rand/1/bin:

$$v_{i,G} = x_{i,G} + F \cdot (x_{r_1,G} - x_{i,G}) + F \cdot (x_{r_2,G} - x_{r_3,G}) \quad (2)$$

DE/best/1/bin:

$$v_{i,G} = x_{best,G} + F \cdot (x_{r_1,G} - x_{r_2,G}) \quad (3)$$

DE/current-to-best/1/bin:

$$v_{i,G} = x_{i,G} + F \cdot (x_{best,G} - x_{i,G}) + F \cdot (x_{r_1,G} - x_{r_2,G}) \quad (4)$$

Where F denotes mutation rate. r_0, r_1, r_2 are different integers which are randomly chosen from the set $\{1, 2, \dots, Np\} \setminus \{i\}$. $x_{best,G}$ denotes the best solution vector at iteration G .

(3) Crossover

A binomial crossover operation generates offspring vector $u_{i,G+1}$. The variables of $u_{i,G+1}$ are calculated as follows:

$$u_{ij,G+1} = \begin{cases} v_{ij,G+1}, & \text{if rand} \leq CR \text{ or } j = j_r \\ x_{ij,G}, & \text{otherwise} \end{cases} \quad (5)$$

Where CR denotes crossover rate. rand is a randomly generated number in the range $[0,1]$. j_r denotes a random integer in the range $[1,N]$.

(4) Selection

For the minimization problem, the better individual has smaller fitness value. Selection operation is to compare the fitness values of the trial individual and parent individual, and the individual with smaller fitness value enters the next generation.

$$x_{i,G+1} = \begin{cases} u_{i,G+1}, & \text{if } f(u_{i,G+1}) \leq f(x_{i,G}) \\ x_{i,G}, & \text{otherwise} \end{cases} \quad (6)$$

(5) Judge terminal condition

If the terminal conditions meet, computation is stopped. Otherwise, mutation, crossover and selection are repeated.

3. Improved Differential Evolution Algorithm with Novel Mutation Strategy

In order to improve the accuracy of DE and jump out of the local optima, scholars have done a lot of improvements. In this paper, DE is improved as follows. Firstly, NMSIDE adopts the novel mutation strategy. Secondly, NMSIDE uses the best solution to help the trapped solutions to escape from local optima. Thirdly, NMSIDE dynamically updates control parameters F and CR .

3.1 Novel Mutation Strategy

When the mutation strategy is DE/current-to-best/1/bin, the best individual is the best individual of the current population, and the two random individuals are randomly selected from the current population, which can cause the individual to fall into the local optima and decrease evolutionary speed. Thus, in order to improve the performance of DE, Zhang et al [7] proposed a new mutation strategy (DE/current-to- $prbest$ /1/bin), which is a semi-greedy strategy. The individual $prbest$ is randomly chosen as one of the top $Np \cdot q\%$ individuals in the current population. The operation diversifies the population. Gong et al [12] proposed a selection mechanism based on the ranking mechanism. The more individual is outstanding, the more the ranking is high. The possibility of being selected is greater. It effectively balances the exploration and exploitation ability of the algorithm.

NMSIDE draws on the experience of scholars. In the mutation operation, NMSIDE chooses a group of individuals randomly in the current population, and selects their best individuals instead of the best individual of the whole population, to some extent, which reduces the greed of the algorithm. Before choosing two random individuals in the perturbation vector, the individuals are sorted according to the fitness values in ascending order. Then, the individuals with good fitness values are divided into the first group, and the individuals with poor fitness are divided into the second group. The number of individuals in the two groups is equal. The individual r_1 is randomly selected from the first group, and the individual r_2 is randomly selected from the second group, which facilitates the evolution of the population in the good direction and accelerates the search speed of the algorithm. The mutation strategy of NMSIDE is as follows:

$$v_{i,G+1} = x_{i,G} + F \cdot (x_{prbest,G} - x_{i,G}) + F \cdot (x_{r_1,G} - x_{r_2,G}) \quad (7)$$

Where $x_{prbest,G}$ denotes the individual $prbest$ at iteration G . The individual $prbest$ randomly selects in $Np \cdot q\%$ individuals, where q takes 50. According to the fitness values in ascending order, the individual population is expressed as $\{1, 2, \dots, Np\}$. The individual r_1 is randomly selected from $\{1, 2, \dots, Np/2\}$. The individual r_2 is randomly selected from $\{Np/2+1, Np/2+2, \dots, Np\}$.

The main purpose of formula (7) is to select the individual $prbest$, r_1 and r_2 , which balances global search and local search of the algorithm.

3.2 Updating the Trapped Solutions

With the evolution of the population, the differences between individuals will become smaller and smaller, which is easy to fall into the local optima, so domestic and foreign scholars do a lot of improvements. Xu et al [13] used two replacement strategies. One replaces stagnant individuals. The other replaces the population of the premature convergence. The two strategies strengthen the exploitation and exploration ability of the algorithm, respectively.

NMSIDE uses the best solution to help the trapped solutions out of the local optima. The expression is as follows:

$$x_{ij,G+1} = \begin{cases} (x_{ij,G+1} + x_{bestj,G}) / 2, & \text{count}(i)=ST \\ x_{ij,G+1}, & \text{otherwise} \end{cases} \quad (8)$$

Where ST denotes stagnation period. $\text{count}(i)$ is used to count the number of continuous stagnation for the i th solution.

The idea of formula (8) is that if the individual search is stagnant, the information of the best individual is used to help the trapped individuals to jump out of the local optima; otherwise the individual remains the same.

3.3 Parameter Adjustment

Control parameters F and CR have an important effect on the performance of the algorithm, where the two parameters are dynamically adjusted. Control parameter F is randomly selected in the range of values. Control parameter CR is incremented based on the number of iterations. The expressions for adjusting parameters F and CR are as follows:

$$F_{i,G+1} = F_{\min} + \text{rand} \cdot (F_{\max} - F_{\min}) \quad (9)$$

$$CR_{i,G+1} = CR_{\min} + (G / G_{\max})^{\frac{1}{4}} \cdot (CR_{\max} - CR_{\min}) \quad (10)$$

Where F_{\min} and F_{\max} denote the minimum and maximum values of F , respectively. $F_{i,G+1}$ represents the mutation rate of the i th individual at iteration G . CR_{\min} and CR_{\max} denote the minimum and maximum values of CR , respectively. $CR_{i,G+1}$ represents the crossover rate of the i th individual at iteration G . G and G_{\max} denote the current number of iterations and the maximum number of iterations, respectively.

3.4 Description of the Proposed Algorithm

Step 1: Initialize the population and parameters, including the population size, the maximum number of iterations, the maximum and minimum of the mutation rate and the crossover rate, the parameter q , the dimension of the individual vector, the boundary of the algorithm search and $G = 1$.

Step 2: Calculate the fitness values of the individual and arrange them in ascending order of fitness values.

Step 3: Implement a novel mutation operation and get the mutated individual v_i .

Step 4: Carry out the crossover operation and get the trial individual u_i .

Step 5: Based on the principle of greed to implement the selection operation, better individual enters the next generation.

Step 6: Updating the trapped solutions according to the best solution.

Step 7: If the maximum number of iterations is researched, computation is stopped. Otherwise, mutation, crossover and selection are repeated.

4. Experimental Results and Analysis

In order to verify the performance of the algorithm, NMSIDE is used to optimize the 11 standard test functions [14,15,16] and compared with several recent typical DE algorithms. The 11 standard test functions are shown in table 1. At the same time, in order to fully explain the influence of the mutation operator on the performance of the algorithm, different mutation operators are compared. Additionally, we use the average number of iterations(AI) and the success rate(SR) [17] to compare the convergence speed and robustness of each algorithm.

Table 1 11 standard test functions

Function	Range	Accuracy	Theory optimal value
$f_1(x) = \sum_{i=1}^N x_i^2$	[-100,100]	1e-50	0
$f_2(x) = \sum_{i=1}^N x_i + \prod_{i=1}^N x_i $	[-10,10]	1e-30	0
$f_3(x) = \sum_{i=1}^N (\sum_{j=1}^i x_j)^2$	[-100,100]	1e-10	0
$f_4(x) = \max \{ x_i , 1 \leq i \leq N\}$	[-100,100]	1e-5	0
$f_5(x) = \sum_{i=1}^N (\lfloor x_i + 0.5 \rfloor)^2$	[-100,100]	1	0
$f_6(x) = \sum_{i=1}^N ix_i^2$	[-5.12,5.12]	1e-60	0
$f_7(x) = \sum_{i=1}^{N-1} [100(x_{i+1} - x_i)^2 + (1 - x_i)^2]$	[-2,2]	1e-8	0
$f_8(x) = \sum_{i=1}^N ix_i^4$	[-1.28,1.28]	1e-100	0
$f_9(x) = (\frac{\pi}{N})(10 \sin^2(\pi y_1) + \sum_{i=1}^{N-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_N - 1)^2)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$	[-10,10]	1e-30	0
$f_{10}(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{N-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_N - 1)^2 [1 + \sin^2(2\pi x_N)])$	[-5,5]	1e-30	0
$f_{11}(x) = \sum_{i=1}^N (10^6)^{\frac{i-1}{N-1}} x_i^2$	[-100,100]	1e-50	0

4.1 Comparison of Algorithm Results

All the experimental results are implemented with Matlab8.3 simulation software, and the experimental computer is configured as Intel (R) Core (TM) i5-2450M CPU @ 2.50GHZ.

NMSIDE is compared with the recent well-known approaches, such as jdDE [18], RMDE [19], HSDE [8] and the basic DE algorithm. The parameters of the algorithm are set as follows. The parameters of jdDE, RMDE and HSDE are consistent with the corresponding literatures. In the basic DE algorithm, F is equal to 0.5 and CR is equal to 0.9. In NMSIDE, q is equal to 50; F_{\max} is equal to 0.9; F_{\min} is equal to 0.2; CR_{\min} is equal to 0.3; CR_{\max} is equal to 0.9; ST is equal to 5. The dimension of all standard test functions is equal to 30, and the population size is equal to 100. For the maximum number of iterations of the test function, the maximum number of iterations of the function f_5 is equal to 100, and the maximum number of iterations of the remaining functions is equal to 1500. Each function runs 30 times independently. The performance of each algorithm is obtained by solving the best value, the mean value, the standard deviation(Std.Dev.), the success rate(SR) and the average number of iterations(AI). SR represents the ratio of the number of runs that meet the convergence accuracy to the total number of runs. AI represents the average number of iterations to achieve convergence accuracy. The experimental results at 30 dimensions are shown in table 2. The best optimization results of NMSIDE are marked in bold font, which suggests that NMSIDE is superior to the other approaches for solving most problems.

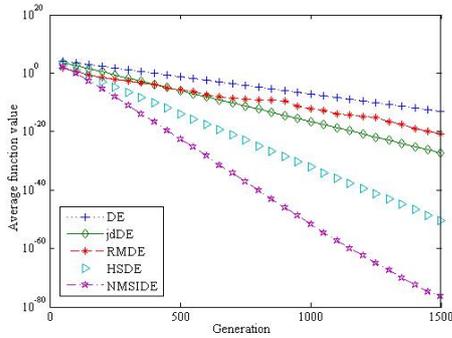
Table 2 30-dimensional simulation results of different algorithms

Function	Algorithm	Best	Mean	Std.Dev.	SR	AI
f_1	DE	9.6222e-15	7.7000e-14	4.1588e-14	0%	1500
	jdDE	1.1891e-28	5.3637e-28	3.8998e-28	0%	1500
	RMDE	8.7695e-29	1.7865e-21	9.5797e-21	0%	1500
	HSDE	2.3989e-53	2.4675e-51	4.8422e-51	100%	1429
	NMSIDE	5.5713e-83	4.0688e-77	1.5516e-76	100%	942
f_2	DE	1.0477e-07	7.7021e-07	4.9265e-07	0%	1500
	jdDE	9.3928e-19	2.0120e-18	6.4658e-19	0%	1500
	RMDE	6.6664e-12	1.8405e-04	9.6004e-04	0%	1500
	HSDE	1.1163e-28	7.3753e-28	7.1988e-28	0%	1500
	NMSIDE	7.3905e-39	7.1489e-32	2.5373e-31	100%	1156
f_3	DE	2.9982e-01	1.1653e+00	7.1738e-01	0%	1500
	jdDE	4.4335e+02	1.0489e+03	6.5779e+02	0%	1500
	RMDE	2.0322e-12	1.5689e-06	8.3433e-06	23%	1467
f_3	HSDE	1.8730e-09	8.1475e-08	1.1823e-07	3%	1500
	NMSIDE	3.7650e-17	1.9633e-11	6.3350e-11	97%	1230
	f_4	DE	4.3457e-03	2.1157e-01	5.1054e-01	0%
jdDE		7.5084e-04	1.5105e-03	4.5392e-04	0%	1500
RMDE		1.3162e-02	1.5408e-01	1.0543e-01	0%	1500
HSDE		1.0089e-06	4.4923e-05	7.4254e-05	93%	1296
NMSIDE		6.9313e-02	1.1644e+00	9.7264e-01	0%	1500
f_5	DE	1.8880e+03	3.5617e+03	8.2326e+02	0%	100
	jdDE	3.0700e+02	4.2220e+02	7.1333e+01	0%	100
	RMDE	0.0000e+00	2.2800e+01	2.8071e+01	43%	79
	HSDE	4.0000e+00	8.3667e+00	2.7226e+00	60%	97
	NMSIDE	0.0000e+00	3.3667e+00	2.6455e+00	100%	69
f_6	DE	1.9445e-16	2.2688e-15	2.0952e-15	0%	1500
	jdDE	3.4222e-30	2.2907e-29	1.7207e-29	0%	1500
	RMDE	2.6088e-32	1.1334e-24	3.0455e-24	0%	1500
	HSDE	1.1194e-55	5.0927e-53	1.4534e-52	0%	1500
	NMSIDE	9.0023e-86	6.6213e-78	2.6715e-77	100%	1098
f_7	DE	1.2432e+01	1.4290e+01	1.1177e+00	0%	1500
	jdDE	1.9533e+01	2.1040e+01	6.3339e-01	0%	1500
	RMDE	2.9033e-11	1.6043e-05	6.5227e-05	73%	1204
	HSDE	2.0468e-01	2.3463e+00	1.3040e+00	0%	1500
	NMSIDE	4.2514e-06	1.8427e+00	2.0050e+00	0%	1500
f_8	DE	1.3517e-30	5.9242e-29	8.7244e-29	0%	1500
	jdDE	4.7543e-49	1.4961e-47	4.0628e-47	0%	1500
	RMDE	2.2659e-54	3.1141e-41	1.5016e-40	0%	1500
	HSDE	4.5120e-84	6.5851e-77	2.5288e-76	0%	1500
	NMSIDE	3.3417e-129	4.4553e-115	2.0652e-114	100%	1132
f_9	DE	6.7900e-18	4.8176e-17	2.8641e-17	0%	1500
	jdDE	3.3776e-32	1.3329e-31	9.5225e-32	100%	1403
	RMDE	1.3058e-31	1.4820e-23	8.1110e-23	23%	1455
	HSDE	1.5705e-32	1.5705e-32	5.5674e-48	100%	813
	NMSIDE	1.5705e-32	1.5705e-32	5.5674e-48	100%	545
	DE	1.1213e-17	4.4687e-17	3.7742e-17	0%	1500

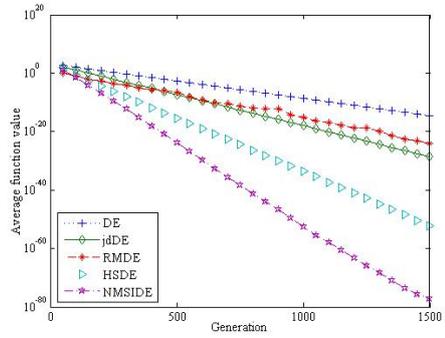
f_{10}	jdDE	5.5406e-32	1.9966e-31	1.6064e-31	100%	1419
	RMDE	2.4591e-32	1.9976e-25	9.1137e-25	47%	1457
	HSDE	1.3498e-32	1.3498e-32	5.5674e-48	100%	1500
	NMSIDE	1.3498e-32	1.3498e-32	5.5674e-48	97%	554
f_{11}	DE	1.0191e-11	5.9732e-11	4.2298e-11	0%	1500
	jdDE	2.2315e-25	1.3689e-24	1.5909e-24	0%	1500
	RMDE	2.2942e-26	9.2855e-17	3.5110e-16	0%	1500
	HSDE	1.4830e-48	5.6437e-47	1.0903e-46	0%	1500
	NMSIDE	4.0102e-78	9.8281e-72	5.3116e-71	100%	1021

As can be seen from table 2, NMSIDE is better than the other four algorithms in the overall performance. Except the functions f_4 and f_7 , the values of NMSIDE are lower than or equal to the other approaches. For the function f_5 , NMSIDE can find the global best solution, which shows that NMSIDE has strong global search ability. For most functions, the mean and standard deviation of NMSIDE is smaller than that of other algorithms, which indicates that NMSIDE has good stability and high convergence precision. For solving the functions $f_1, f_2, f_3, f_5, f_6, f_8, f_9$ and f_{11} , the success rate of NMSIDE is higher than that of the other algorithms. For solving the functions $f_1, f_2, f_5, f_6, f_8, f_9$ and f_{11} , the success rate of NMSIDE is 100%, which indicates that NMSIDE has strong robustness. For any function which fail to converge to the global optima, the success rate is 0, and AI is not discussed below. For the successful convergence functions, the average number of iterations of NMSIDE is less than or equal to the other algorithms, which shows that NMSIDE has a fast convergence rate.

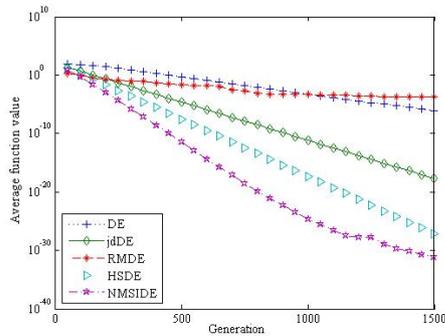
In order to visualize the optimization performance of NMSIDE, in the 30-dimensional, convergence curves for four functions of all test functions are shown in Fig. 2. The horizontal axis is the number of generations, and the vertical axis is average function value over 30 independent runs.



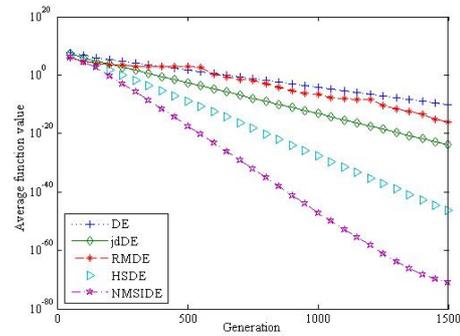
(a) Function f_1



(c) Function f_6



(b) Function f_2



(d) Function f_{11}

Fig. 2 Convergence curves for four test functions

Convergence curves for four test functions can be obviously seen that NMSIDE convergence speed is more fast than the other state-of-the-art algorithms.

4.2 Operator Analysis of NMSIDE

In order to further analyze the effects of the mutation operator on NMSIDE, NMSIDE is compared with the different mutation operators (DE/rand/1, DE/best/1, DE/current-to-rand/1, DE/current-to-best/1). The performance differences between the different operators are obtained by comparing the mean fitness value and the standard deviation. The parameter settings of each operator are the same as that of 30-dimensional algorithm comparisons. The comparative analysis of the five operators is shown in table 3 below. The best optimization results of NMSIDE are marked in bold font, which suggests that the operator of NMSIDE is superior to the other operators for solving most problems.

Table 3 Comparison of mutation operator performance

Function/ Operator	rand/1	best/1	current-to-rand/1	current-to-best/1	NMSIDE
	Mean (Std. Dev.)				
f_1	4.9330e-39 (8.4878e-39)	1.5849e-05 (8.6808e-05)	2.2870e-25 (4.3085e-25)	3.9324e-53 (2.1534e-52)	4.0688e-77 (1.5516e-76)
f_2	6.6667e-23 (2.9193e-23)	3.0224e-06 (1.1484e-05)	3.6286e-19 (1.9788e-18)	1.0267e-24 (5.6233e-24)	7.1489e-32 (2.5373e-31)
f_3	7.4807e-01 (1.5108e+00)	5.7973e+01 (2.7635e+02)	3.3357e+01 (2.3734e+01)	1.2744e-06 (4.5712e-06)	1.9633e-11 (6.3350e-11)
f_4	3.9799e+00 (2.0152e+00)	1.8930e+01 (5.4140e+00)	2.7203e+00 (9.0264e-01)	4.6576e+00 (1.8182e+00)	1.1644e+00 (9.7264e-01)
f_5	0.0000e+00 (0.0000e+00)	5.4000e+00 (7.0496e+00)	0.0000e+00 (0.0000e+00)	2.3333e-01 (5.0401e-01)	3.3667e+00 (2.6455e+00)
f_6	9.3375e-41 (1.4522e-40)	1.2151e-12 (6.6556e-12)	5.4627e-26 (2.0944e-25)	2.0035e-56 (1.0605e-55)	6.6213e-78 (2.6715e-77)
f_7	2.4518e+01 (2.7875e-01)	2.5614e+01 (1.5700e+01)	2.7106e+01 (5.0230e-01)	7.4482e+00 (4.2600e+00)	1.8427e+00 (2.0050e+00)
f_8	1.9098e-34 (1.0460e-33)	7.7617e-14 (4.2439e-13)	2.0777e-14 (1.0885e-13)	3.5587e-80 (1.9293e-79)	4.4553e-115 (2.0652e-114)
f_9	3.4556e-03 (1.8927e-02)	3.8011e-02 (6.3747e-02)	3.2969e-26 (1.7963e-25)	3.4556e-03 (1.8927e-02)	1.5705e-32 (5.5674e-48)
f_{10}	1.3498e-32 (5.5674e-48)	7.3249e-04 (2.7876e-03)	3.3807e-26 (1.7493e-25)	1.3498e-32 (5.5674e-48)	1.3498e-32 (5.5674e-48)
f_{11}	6.6378e-35 (2.6979e-34)	8.6431e+02 (4.7340e+03)	3.3332e-21 (1.3799e-20)	6.4230e-44 (3.5180e-43)	9.8281e-72 (5.3116e-71)

As can be seen from table 3, the operators DE/rand/1, DE/best/1, DE/current-to-rand/1 and DE/current-to-best/1 optimize many functions that the average and standard deviation are large, which indicates that these four operators under the framework of NMSIDE algorithm are easy to fall into the local optima, and the convergence accuracy and stability are low. For most functions, the average and standard deviation of NMSIDE are less than or equal to the average and standard deviation of the other four operators, which indicates that the convergence accuracy and stability of NMSIDE are higher.

5. Conclusions

In order to improve the deficiency of differential evolution algorithm, an improved differential evolution algorithm with novel mutation strategy is presented. In the novel mutation strategy, the individual *prbest* is chosen to replace the original best individual, which reduces the greed of the algorithm, and makes the target vector move in the direction of good direction. The trapped solutions are updated according to the best solution. NMSIDE dynamically updates the control parameters to meet the requirements of the algorithm for different search phases. The experimental results show that NMSIDE has good performance. Although NMSIDE shows some advantages for most functions, some parameters in NMSIDE are based on the actual experience, and there is no theoretical guidance. NMSIDE has poor convergence performance for minority functions, which indicates that the algorithm has limitation. The future will strengthen the theoretical research of algorithm parameters and improve the practicality of the algorithm.

6. Acknowledgments

This work was financially supported by the National Natural Science Foundation of China (No. 61403174) and Jiangsu Province Graduate Scientific Research Innovation Project(KYCX17_1575).

References

- [1] R. Storn, K. Price. Differential Evolution-a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces[J]. *Journal of Global Optimization*,1997,11(4):341-359.
- [2] D. H. He, Y. X. Li ng, W. Y. Gong, et al. An Adaptive Differential Evolution Algorithm for Constrained Optimization Problems[J]. *Acta Electronic Sinica*,2016,44(10):2535-2542.
- [3] A. J. Li, J. Wang, J. Li, et al. Clearance of Flight Control Law Based on Cultural Differential Evolution Algorithm[J]. *Pattern Recognition and Artificial Intelligence*,2014,27(3):256-262.
- [4] D. X. Zou, S. Li, G. G. Wang, et al. An Improved Differential Evolution Algorithm for the Economic Load Dispatch Problems with or without Valve-Point Effects[J]. *Applied Energy*,2016,181:375-390.
- [5] J. Kennedy, R. C. Eberhart. Particle Swarm Algorithm[C]. *Proceedings of the 1995 IEEE International Conference on Neural Networks*. New York: IEEE Press,1995,4:1942-1948.
- [6] D. Karaboga, B. Akay. A Comparative Study of Artificial Bee Colony Algorithm[J]. *Applied Mathematics & Computation*,2009,214(1):108-132.
- [7] J. Q. Zhang, A. C. Sanderson. JADE: Adaptive Differential Evolution with Optional External Archive[J]. *IEEE Transaction on Evolutionary Computation*,2009,13(5):945-957.
- [8] W. C. Yi, L. Gao, X. Y. Li, et al. A New Differential Evolution Algorithm with a Hybrid Mutation Operator and Self-Adapting Control Parameters for Global Optimization Problems[J]. *Applied Intelligence*,2015,42(4):642-660.
- [9] S. M. Islam, S. Das, S. Ghosh, et al. An Adaptive Differential Evolution Algorithm with Novel Mutation and Crossover Strategies for Global Numerical Optimization[J]. *IEEE Transactions on Systems Man & Cybernetics Part B*,2012,42(2):482-500.
- [10] Y. Z. Zhou, W. C. Yi, L. Gao, et al. Adaptive Differential Evolution With Sorting Crossover Rate for Continuous Optimization Problems[J]. *IEEE Transactions on Cybernetics*,2017,PP(99):1-12.
- [11] D. X. Zou, L. Q. Gao, J. H. Wu, et al. Application of HDEHS Algorithm to Structural Engineering[J]. *Journal of Northeastern University: Natural Science*,2010,31(6):769-772.
- [12] W. Y. Gong, Z. H. Cai. Differential Evolution with Ranking-Based Mutation Operators[J]. *IEEE Transaction on Cybernetics*,2013,43(6):2066-2081.
- [13] C. Xu, H. Huang, S. Ye. A Differential Evolution with Replacement Strategy for

Real-Parameter Numerical Optimization[C]// Evolutionary Computation. IEEE, 2014:1617-1624.

- [14] S. Rahnamayan, H. R. Tizhoosh, M. M. A. Salama. Opposition-Based Differential Evolution[J]. IEEE Transaction on Evolutionary Computation,2008,12(1):64-79.
- [15] M. M. Ali, C. Khompatraporn, Z. B. Zabinsky. A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems[J]. Journal of Global Optimization,2005,31(4):635-672.
- [16] S. H. Wang, Y. Z. Li, H. Y. Yang, et al. Self-Adaptive Differential Evolution Algorithm with Improved Mutation Strategy[J]. Soft Computing,2017:1-15.
- [17] J. F. Qiao, S. P. Fu, H. G. Han. A Modified Differential Evolution Algorithm Based on Hybrid Mutation Strategy for Function Optimization[J]. Control Engineering,2013,20(5):943-947.
- [18] L. Y. Jia, J. X. He, C. Zhang, et al. Differential Evolution with Controlled Search Direction[J]. Journal of Central South University,2012,19(12):3516-3523.
- [19] H. B. Ouyang, L. Q. Gao, X. Y. Kong. Random Mutation Differential Evolution Algorithm[J]. Journal of Northeastern University: Natural Science,2013,34(3):330-334.