

An Improved Adaptive Niche Differential Evolution Algorithm

Hui Wang*, Changtong Song

Electron & Information Department, Zhenjiang College, Zhenjiang 212000, China

*Corresponding author, e-mail: fl@dnzs.net.cn

Abstract

Differential evolution (DE) algorithm is a random search algorithm by referring to the natural genetic and natural selection mechanism of the biological world and it is used to process the complicated non-linear problems which are difficult to be solved by traditional computational methods. However, subject to its own mechanism and single structure, the basic DE algorithm is easy to get trapped into local optimum and it is difficult to handle high-dimensional and complicated optimization problems. In order to enhance the search performance of the DE algorithm, this paper uses the idea of niche, decomposes the entire population into several niches according to the fitness, perform population selection by integrating the optimum reservation strategy to realize the optimal selection of niche, adjusts the fitness of the individual of the population, designs the adaptive crossover and mutation operators to make the crossover and mutation probabilities change with the individual fitness and enhances the ability of DE algorithm to jump out of the local optimal solution. The experiment result of benchmark function shows that the method of this paper can maintain solution diversity, effectively avoid premature convergence and enhance the global search ability of DE algorithm.

Keywords: differential evolution, niche algorithm, adaptive crossover, adaptive mutation

Copyright © 2016 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

In the field of information science, evolutionary computation, which is affected by the natural selection mechanism of “survival of the fittest” and the transmission rules of genetic information, takes the problems to be solved as the environment and searches the optimal solution through natural evolution from the population formed by the possible solutions. As a universal optimization algorithm based on natural selection and genetic theory, DE algorithm has been successfully applied in many fields. Although DE algorithm has many advantages in solving the optimization problems, there is still some room to be improved. For example, it fails to maintain population diversity and it is easy to get trapped into local extreme points of multi-peak function [1]. In a word, DE algorithm is not maturely developed yet; therefore, continuous research is required so as to expand the application fields of DE algorithm. As an effective approach to solve multi-peak optimization problems, niche has drawn extensive attention and it has become a research focus in DE algorithm. Niche method can reduce the disturbance and the combination of DE algorithm can make up for the defects of DE algorithm in solving local extremum and it has certain advantages in solving multi-modal, highly-dimensional, multi-objective and dynamic complicated optimization problems [2, 3].

Based on population difference, DE algorithm was proposed by Rainer Storn and Kenneth Price in the year of 1996 and its basic idea is to obtain the interim population by reorganizing the differences of individuals of the current population and obtains a new generation of population through the competition of offspring individuals and parent individuals. In the DE algorithm, the mutated individuals are formed through the mutation operation of the parent individuals, then, it performs crossover operation between the parent individuals and the mutated individuals based on certain probability and produces test individuals. After that, it conducts greedy selection operation on the parent individuals and the test individuals according to the fitness, retains the better individuals and realizes the population evolution. However, DE algorithm has such problems as bad local search ability, low search efficiency in the post evolution phase and premature convergence, which selection methods to use has been a difficulty for DE algorithm all the time so as to retain the excellent individuals and maintain the

population diversity [4]. Niche technology has been proposed in the niche implementation method based on pre-selection mechanism by Cavichio in the 1970s for the first time. The creature in the natural world lives with the individuals and population with similar shapes and features to its own, including selecting mates and producing offspring, under such natural rules and laws, the natural world gradually develops and enriches. Niche technology decomposes the genetic individuals of each generation into several kinds, selects certain individuals with bigger fitness from each kind as the excellent representatives to form a population and produces a new population within the population and among different populations through crossover and mutation [5].

This paper firstly analyzes the characteristics of such operations as crossover, mutation and selection of DE algorithm. Then, it introduces niche technology into DE algorithm, performs structural design of niche differential algorithm, uses adaptive probability strategy in the crossover and mutation of DE algorithm, ensures accelerated evolutionary optimization of the entire population in the early evolution phase and avoids damage on the population optimization in the post evolution so as to maintain the population diversity, retains certain distance between the individuals, create a niche evolutionary environment and improve the convergence speed and the global search performance of DE algorithm. The final experiment result proves that the method of this paper has stable convergence performance and higher computation efficiency in solving complicated optimization problems.

2. Operations of Differential Evolution Algorithm

Assume that the current evolution generation is t , the population size is NP , the space dimension is D , the current population is $X(t) = \{x_1^t, x_2^t, \dots, x_{NP}^t\}$ and $x_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t)$ is the i th individual of the population. Perform the following three operations on each individual x_i^t successively.

(1) Mutation

DE algorithm adds the weighted difference vector between the two members of the population to the third member to produce new parameter vector and this operation is called mutation. Every individual x_i^t will produce the mutation individual $v_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{iD}^t)$ according to the formula below [6].

$$v_{ij}^t = x_{ij}^t + \lambda(x_{r_2j}^t - x_{r_3j}^t) \quad j = 1, 2, \dots, D \quad (1)$$

In this formula, $x_{r_1}^t = (x_{r_11}^t, x_{r_12}^t, \dots, x_{r_1D}^t)$, $x_{r_2}^t = (x_{r_21}^t, x_{r_22}^t, \dots, x_{r_2D}^t)$ and $x_{r_3}^t = (x_{r_31}^t, x_{r_32}^t, \dots, x_{r_3D}^t)$ are three individuals randomly selected from the population and $r_1 \neq r_2 \neq r_3 \neq i$, λ is a scaling factor with a range of $[0, 2]$.

(2) Crossover

DE algorithm mixes the parameter of the mutation vector and the predefined target parameter and produces the test vector according to certain rules, known as crossover. The test individual $u_i^t = (u_{i1}^t, u_{i2}^t, \dots, u_{iD}^t)$ can be produced according to the mutation individual v_i^t and the parent individual x_i^t and

$$u_{ij}^t = \begin{cases} v_{ij}^t & \text{if } rand \leq CR \text{ or } j = rand_j \\ x_{ij}^t & \text{if } rand > CR \text{ and } j \neq rand_j \end{cases} \quad (2)$$

In this formula, $rand$ is a random number within the scope of $[0, 1]$, CR is a constant within $[0, 1]$, which is called crossover factor and $rand_j$ is a random integer within $[1, D]$ [7].

(3) Selection

If the cost function of the test vector is lower than that of the target vector, the test vector will replace the target vector in the next generation. The final operation is called

selection, which is to select the individual with the best fitness from the parent individual x_i^t and the test individual u_i^t as the individual x_i^{t+1} of the next generation [8].

$$x_i^{t+1} = \begin{cases} x_i^t & \text{if } \text{fit}(x_i^t) < \text{fit}(u_i^t) \\ u_i^t & \text{otherwise} \end{cases} \quad (3)$$

In this formula (3), $\text{fit}(\cdot)$ is the fitness function.

Figure 1 is the schematic for DE algorithm to perform optimization search on multi-peak function.

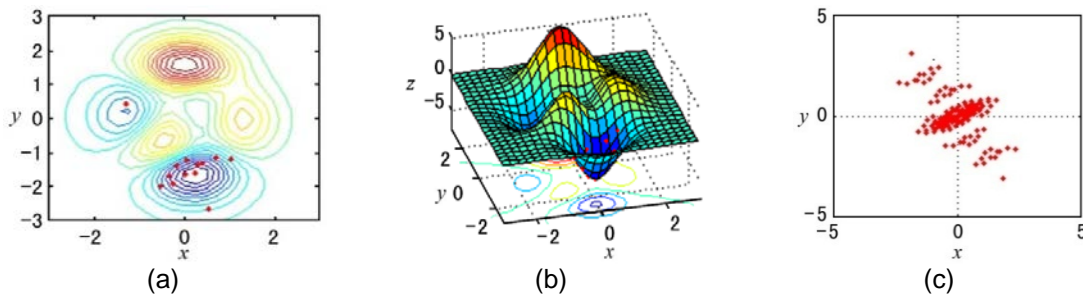


Figure 1. Optimization search of differential evolution algorithm on multi-peak function

3. Design of Adaptive Niche Differential Evolution Algorithm

The basic idea of adaptive niche differential evolution algorithm is to adjust the fitness of each individual from the population by reflecting the similarities of these individuals, based on which the algorithm can perform selection operation and realize the evolution operation environment of niche. The radius of the niche is of great importance. If it is too small, there may be too many niches while if it is too big, many small niches can be seen as one niche and it will affect the evolution process of the niche [9]. The Hamming distance of any two individuals X_i and X_j in a the population is shown as:

$$d_{ij} = \|X_i - X_j\| = \sqrt{\sum_{k=1}^M (x_{ik} - x_{jk})^2} \quad (i = 1, 2, \dots, M + N - 1, j = i + 1, \dots, M + N) \quad (4)$$

X_i and X_j are the i th and j th individuals respectively and N is the number of the initial populations.

Usually, a function which indicates the relationship degree of distance of two individuals in the population is called sharing function and it is marked as $P(i, j)$.

$$P(i, j) = 1 - d(i, j) / p \quad (5)$$

In this formula, d_{ij} is the relationship of the distance between individual i and individual j .

Sharing degree is a measurement to measure the degree of sharing of a certain individual in the population and it is defined as the sum of the sharing functions of this individual and other individuals of the population. It is demonstrated as follows with S_i .

$$S_i = \sum_{j=1}^N S(d_{ij}), i = 1, 2, \dots, N \quad (6)$$

In this formula, N is the population size.

Assume that the fitness of the individual i is F_i and the population size is N , then the probability P_i for individual i to be selected is:

$$P_i = F_i / \left(\sum_{i=1}^N F_i \right) \quad (i = 1, 2, \dots, N) \quad (7)$$

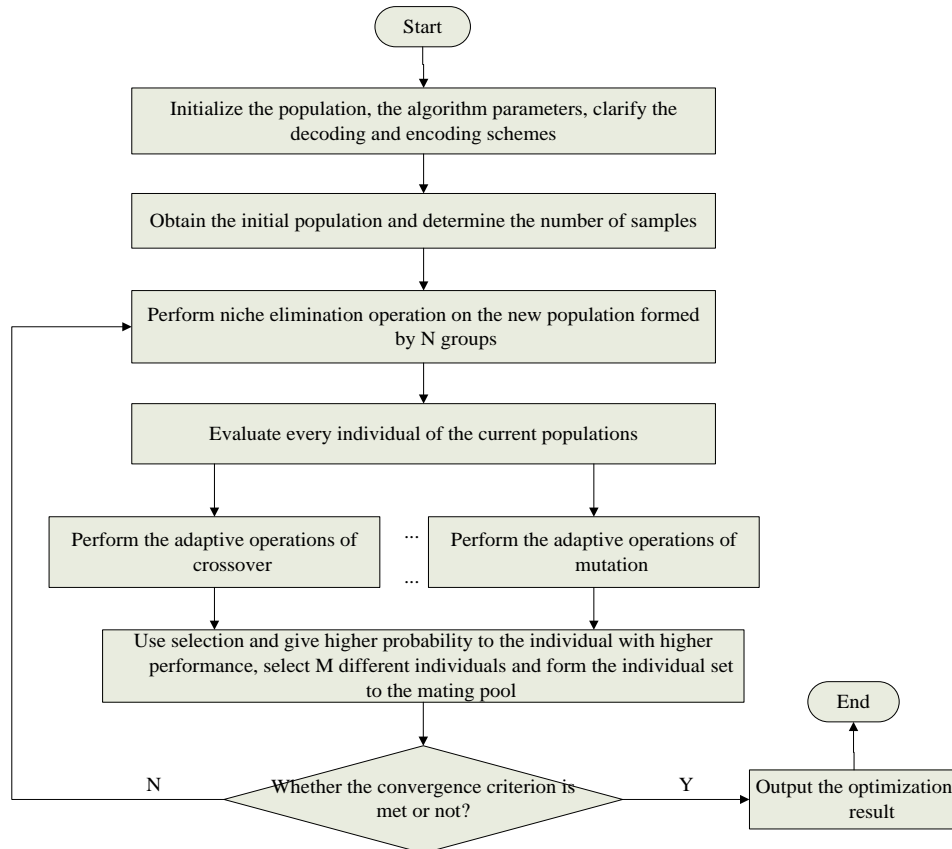


Figure 2. Flowchart of optimization operations of adaptive niche differential evolution algorithm

The steps of adaptive niche differential evolution algorithm are clarified as follows:

(1) Set the evolution generation counter $t=1$, randomly produce N initial individuals and form the initial population $P(t)$, initialize the crossover probability P_c and the mutation probability P_m .

(2) Obtain the fitness of each individual $F_i(t) (i=1, 2, \dots, N)$ and maintain the individual X_{\max} with the maximum fitness.

(3) Adjust P_c and P_m according to Formulas (4)-(6).

Randomly select two individuals from the parent population of each niche. Retain the individuals with big fitness, in order to ensure the crossover quality, the crossover probability is produced in the adaptive way and the adaptive crossover probability is determined by the following formula.

$$P_c = \begin{cases} k_1 (f_{\max} - f_c) / (f_{\max} - f_a) & f_c \geq f_a \\ k_1 & f_c < f_a \end{cases} \quad k_1 \in (0, 1] \quad (8)$$

In this formula, f_{\max} and f_a are the maximum fitness and the average fitness of the parent generation respectively and f_c is the bigger fitness of two individuals to be cross-overed.

Non-uniform mutation operation is used and the mutation probability is controlled by the evolution degree in order to guarantee the mutation quality. The adaptive mutation probability is determined by the formula below.

$$P_m = \begin{cases} k_2(f_{\max} - f_m) / (f_{\max} - f_a) & f_m \geq f_a \\ k_2 & f_m < f_a \end{cases} \quad k_2 \in (0,1] \quad (9)$$

In this formula, f_{\max} and f_a are the maximum fitness and the average fitness of the parent population respectively and f_m is the fitness of the individual to be mutated.

(4) Selection operation. Use selection and give higher probability to the individual with higher performance, select M different individuals according to Formula (7) and form the individual set to the mating pool and obtain the new population.

(5) Convergence criterion judgment. If the convergence criterion is not met, update the counter $t=t+1$ and retain the best individual directly into the next generation, obtain the fitness $fit_i(t)$ of each individual, retain the current optimal individual $X_{\max}(t)$ and turn to Step 3, otherwise, output the optimal layout result.

Based on the above analysis, Figure 2 is the flowchart of the optimization operations of adaptive niche differential evolution algorithm.

4. Experiment Simulation and Analysis

4.1. Test Function

In order to test the performances of ANDE such as the ability to overcome premature convergence and the global optimization capability, four functions showed by Formula (10) to Formula (13) have been selected as the test functions. Among them, f_1 is a single-peak function while $f_2 - f_4$ are multi-peak functions. See their respective characteristics in Table 1.

$$f_1(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \quad (10)$$

$$f_2(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1] \quad (11)$$

$$f_3(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) + 20 + e \quad (12)$$

$$f_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10] \quad (13)$$

Table 1. Characteristics of test functions

Function	Variable range	Variable length	Global optimal solution
f_1	$[-100,100]^n$	8	0
f_2	$[-1.28,1.28]^n$	10	0
f_3	$[-32,32]^n$	10	0
f_4	$[-5.12, 5.12]^n$	12	0

f_1 is a continuous single-peak function. Surrounding the valley is the relatively smooth surface and it is mainly used to test the optimization accuracy of the algorithm.

f_2 has many unevenly distributed peaks, which have different lengths. It has several local maximum and minimum and a huge oscillation.

f_3 has a multi-peak and multi-valley surface and it has one and only one optimal extremum. Besides, it has the spatial distribution of different heights and peaks and it is usually used to measure the performance of the search algorithm in processing the optimization problems with many noises.

f_4 has a strong oscillation as well as many traps. Due to the oscillation and the many local optimal points surrounding the global optimal point, it can trick and misguide the population search into local optimal points and it is quite deceptive to the algorithm.

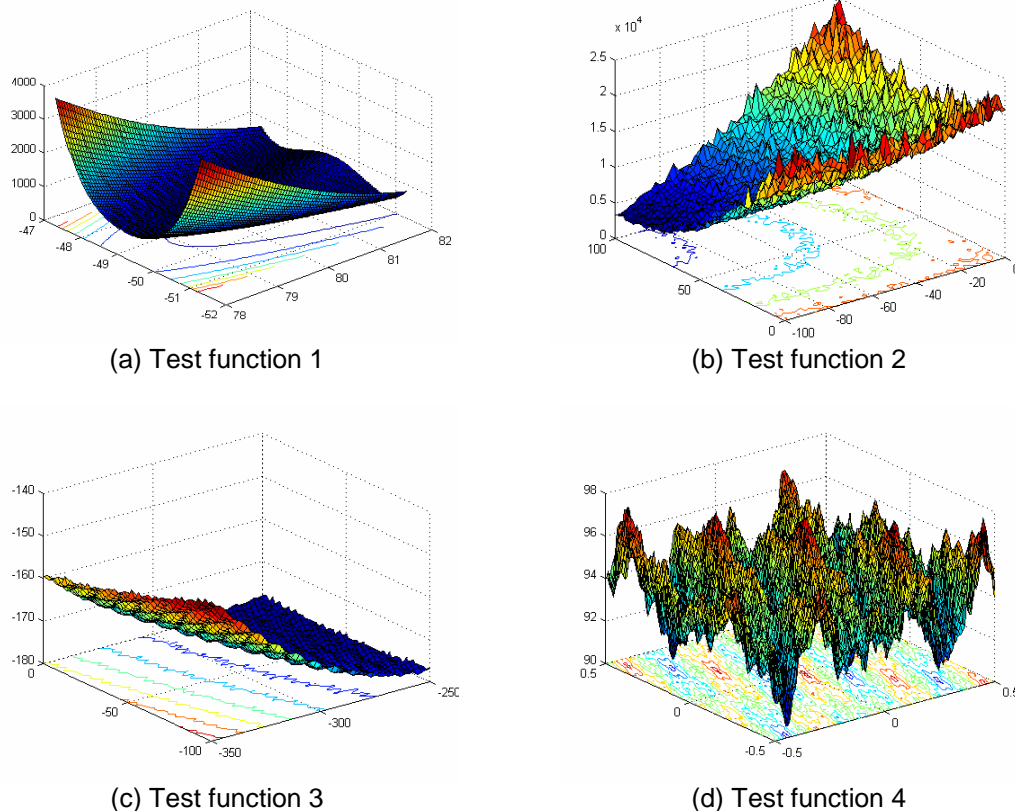


Figure 3. Three-dimensional surfaces of four test functions

From the above analysis, it can be seen that the four functions of $f_1 - f_4$ are generally representative and they can be used to test the optimization performance of the algorithm.

4.2. Comparison of global optimization performance

Table 2 is the statistic comparison of the global optimization performance after repeating 30 times on every test function with ANDE and BDE when the maximum iteration is 500. *best*, *worst*, *mean* and *std* are the optimal solution, the worst solution, the mean value and the mean square error. Table 2 is the statistic comparison of the global optimization performance.

Table 2. Statistic comparison of the global optimization performance

Function	Algorithm	Best	Worst	Mean	Std
f_1	BDE	0.9515×10^{-4}	3.6×10^{-4}	1.8271×10^{-4}	8.7625×10^{-5}
	ANDE	1.3826×10^{-5}	2.3×10^{-4}	6.5533×10^{-5}	6.6382×10^{-5}
f_2	BDE	1.2×10^{-5}	0.0822	0.0232	0.0247
	ANDE	1.0167×10^{-5}	5.85×10^{-4}	1.9288×10^{-5}	1.1636×10^{-4}
f_3	BDE	3.2774	7.7320	5.3260	0.6552
	ANDE	2.5453	5.0036	3.8558	0.8749
f_4	BDE	1.8966	8.9963	4.8756	0.5466
	ANDE	2.9628	8.2558	5.1885	0.5073

It can be seen from the data in Table 2 that ANDE has the best optimization performance while BDE has the worst and that ANDE can jump out of the local optimum more effectively, reduce the premature convergence and show strong optimization performance.

As for f_1 , ANDE has similar performance with BDE, however, BDE has dropped into the deep pit surrounding the global optimum in 30 experiments. Therefore, compared with BDE, ANDE has better performance to overcome local optimization.

For f_2 and f_3 , the average iterations of ANDE are bigger than those of BDE because f_2 and f_3 have plenty of local optimal points and ANDE can jump out of the local optimization, leading to the increase of average iterations. Besides, it can also be seen in the experiment that when the cyclic iterations increase, ANDE can have more global optimal solutions but BDE barely changes.

5. Conclusion

Based on the shortcomings and flaws of DE algorithm in theory and application technology, this paper has introduced niche and sharing degree in the optimization computation. Besides, in the evolution process, it has limited the increase of other individuals by adjusting the fitness of each individual and created niche evolution environment. In the meanwhile, it has used crossover and mutation operators, which are good for the diversity of the population evolution, and improved the search algorithm of the algorithm. The experiment shows that the algorithm proposed by this paper effectively maintains population diversity, has faster convergence rate, jumps out from the local optimum, alleviates prematurity and shows stronger optimization performance.

References

- [1] Lingjuan Hou, Zhijiang Hou. A novel discrete differential evolution algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(4): 1883-1888.
- [2] Dexuan Zou, Jianhua Wu, Liqun Gao, Steven Li. A Modified Differential Evolution Algorithm for Unconstrained Optimization Problems. *Neurocomputing*. 2013; 120(23) :469-481.
- [3] Camila Silva de Magalhães, Diogo Marinho Almeida, Helio José Correa Barbosa, Laurent Emmanuel Dardenne. A Dynamic Niche Genetic Algorithm Strategy for Docking Highly Flexible Ligands. *Information Sciences*. 2014; 289(24): 206-224.
- [4] Linda Slimani, Tarek Bouktir, Algeria. Optimal Power Flow Solution of the Algerian Electrical Network using Differential Evolution Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(2): 199-210.
- [5] Mostafa Z Ali, Noor H Awad. A Novel Class of Niche Hybrid Cultural Algorithms for Continuous Engineering Optimization. *Information Sciences*. 2014; 267(20): 158-190.
- [6] Mohammad H Moradi, Mohammad Abedini, SM Reza Tousi, S Mahdi Hosseinian. Optimal Siting and Sizing of Renewable Energy Sources and Charging Stations Simultaneously Based on Differential Evolution Algorithm. *International Journal of Electrical Power & Energy Systems*. 2015; 73(12): 1015-1024.
- [7] Martin Kotyrba, Eva Volna, Petr Bujok. Unconventional Modelling of Complex System via Cellular Automata and Differential Evolution. *Swarm and Evolutionary Computation*. 2015; 25(12): 52-62.
- [8] Carlos Segura, Carlos A Coello Coello, Alfredo G Hernández-Díaz. Improving The Vector Generation Strategy of Differential Evolution for Large-Scale Optimization. *Information Sciences*. 2015; 323(1): 106-129.
- [9] Javier E Vitela, Octavio Castaños. A Sequential Niche Memetic Algorithm for Continuous Multimodal Function Optimization. *Applied Mathematics and Computation*. 2012; 218(17): 8242-8259.