Solving CEC 2015 Multi-modal Competition Problems Using Neighborhood Based Speciation Differential Evolution

B. Y. Qu[1,2], J. J. Liang[1], Z. Y. Wang[1]
School of Information Engineering
Zhengzhou University
Zhengzhou, China
qby1984@hotmail.com

D. M. Liu[2]
School of Electrical and Information Engineering
Zhongyuan University of Technology
Zhengzhou, China

***Abstract*—In this article, a recently proposed niching algorithm called neighborhood based speciation Differential Evolution (NSDE) is used to solve CEC 2015 multi-modal competition problems. Although DE algorithm is effective in solving single global optimal, the result is not acceptalbe when solving multi-optima problems. NSDE was proposed to enable DE with the ability of handling multi-modal optimization problems. In NSDE, the mutation is performed within each Euclidean neighborhood. During the evolution the population of NSDE will evolve toward the respective global/local optimum and the neighborhood mutation can maintain the multiple optima found. The performance of NSDE is compared with the original SDE. From the simulation results, we can observe that NSDE is effective in solving multi-modal optimization problems.**

Keywords—differential evolution; multimodal optimization; neighborhood mutation, niching algorithm

I. INTRODUCTION

In real world application, many optimization problems contain multiple high quality global or local solutions. For this kind of problems, it is generally desirable to find multiple optima solutions and the most appropriate solution can be selected according to user's prime target [1]. These problems are indentified as multi-modal optimization problems. Multi-modal problems are generally more difficult than single global optima problems as the search resources have to been distributed into difficult areas that may potentially contain a global/local optima.

To solve multimodal optimization problem, niching techniques are commonly used [2],[3]. Different niching methods have been developed in past few years such as crowding [4], fitness sharing [5], restricted tournament selection [6] and speciation [7], [8]. The crowding method was firstly introduced by De Jong in 1975 [4] and it is one of the simplest niching method to solve multimodal optimization problems. In this method, the offspring produced in each generation is compared with several randomly picked individuals from the current population. In this way, the method is able to maintain the diversity of the population as well as the peaks located. Although the concept of Crowding is simple, it is very effective in solving multi-modal optimization problems. The basic idea of Restricted Tournament Selection is similar to crowding. It chooses a random sample of ‘w’ (window size) individuals from the population and the generated offspring will compare with the closest (measured by Euclidean distance) individual within this sample. The fitness sharing was first introduced by Holland and extended by Goldberg and Richardson [5]. The method divide the population into different subgroups according to the similarity of the individual within the current population. The members share their information with each other inside the subgroup. Speciation is another commonly used niching technique in solving multi-modal optimization [8]. The method divide the population into different species according to their similarity. The center of a species is named as species seed. In each generation, the species seed is identified first and the members of each species are identified accordingly [8].

Using Evolutionary Algorithms (EAs) to solve multi-modal optimization problems is a trend in recent year. Differential Evolution is one of the most popular evolutionary algorithms. It was proposed by Storn and Price in 1995 [9]. The original form of DE is used to handle single global optimization problems. Various niching techniques have been incorporated into DE in order to enable it with the ability of solving multi-modal optimization problems [1]. In this paper, a recently proposed DE based niching algorithm called Neighborhood based Speciation Differential Evolution (NSDE) is used to solve the CEC 2015 multi-modal competition problems. The results are compared with the original SDE. As can be seen from the result, NSDE is able to generate better results.

The remainder of this article is organized as follows. Section II introduce differential evolution (DE) algorithm. Section III presents the species differential evolution as well as the neighborhood based species differential evolution. Experimental Preparations is introduced in Section IV. Section V and VI gives experimental results and conclusion respectively.

This research is partially supported by National Natural Science Foundation of China (61305080, 61473266, 61379113) and Postdoctoral Science Foundation of China (2014M552013).

978-1-4799-7492-4/15/$31.00 ©2015 IEEE

II. DIFFERENTIAL EVOLUTION

DE is frequently used in global optimization and different from other algorithms. Although the idea of DE is simple , it is identified as one of the most powerful EAs [10]. The flowchart of DE algorithm is presented in Fig. 1. There are three important operations can be identified in DE known as: mutation, crossover and selection.

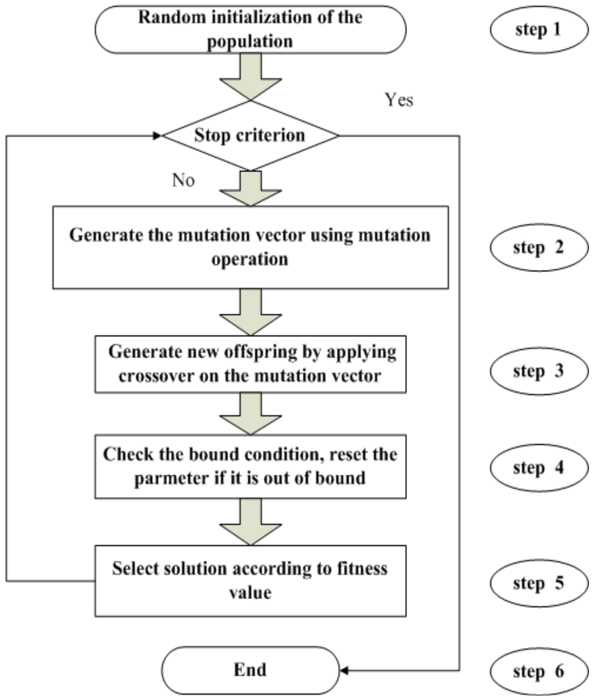


Fig. 1. The flowchart of DE algorithm

- Mutation: DE algorithm generally produces the mutation vector v_p associated with the parent vector x_p by using the following equations [11]:

  DE/rand/1: $v_p = x_{r1} + F \bullet (x_{r2} - x_{r3})$

  DE/best/1: $v_p = x_{best} + F \bullet (x_{r2} - x_{r3})$

  DE/current-to-best/2:
  $v_p = x_p + F \bullet (x_{best} - x_p) + F \bullet (x_{r1} - x_{r2})$

  DE/best/2:
  $v_p = x_{best} + F \bullet (x_{r1} - x_{r2}) + F \bullet (x_{r2} - x_{r4})$

  DE/rand/2:
  $v_p = x_{r1} + F \bullet (x_{r2} - x_{r3}) + F \bullet (x_{r4} - x_{r5})$

  where r_1, r_2, r_3, r_4, r_5 are random and mutually different integers. F is the scaling factor generally in the range [0, 1] and it is used to scale the differential vector.

- Crossover: The crossover operation is applied to the newly generated mutant vector and the corresponding parent vector after the mutation operation. This operation is used to increase the diversity of population. It can be expressed as:

$$u_{p,i} = \begin{cases} v_{p,i}, & \text{if } rand_j \le CR \text{ or } j=k \\ x_{p,i}, & \text{otherwise} \end{cases}$$

  Where u_p is the newly generated offspring vector. CR is a user-specified crossover rate and the value is in the range [0,1). It determines the probability of crossover operation for each dimension.

- Selection: The selection operation is last operation of DE algorithms. It generally uses one to one selection method, i.e., the newly generated offspring will compare with its own parent and the fitter one will survive for next generation.

III. SDE AND NSDE

A. Species DE

SDE is a commonly used niching algorithm which is first proposed by Li [8]. The standard speciation technique is adopted in SDE to divide the DE population into different niches. In each generation, the species need to be identified. The steps of identifying the species are listed in Table I and the SDE algorithm are presented in Table II [12].

TABLE I. THE STEPS OF IDENTIFYING THE SPECIES

Input	Sorted population
	While the sorted population is not empty
Step 1	Set the best unprocessed individual in the sorted population as the new species seed.
Step 2	For all the rest unprocessed individuals.
Step 3	If the distance between unprocessed solution and the new species seed is less than the user defined radius (*Rs*), the member will be marked as processed and put into this new species.
	Endfor
Step 4	Remove this species seed as well as processed Individuals from the sorted population.
	Endwhile
Output	A listed of species and their species seed

TABLE II. THE SDE ALGORITHM

Step 1	Randomly initialize the population
Step 2	Evaluate all members in the population using the objective function.
Step 3	Sort all members in descending order according to the fitness Values.
Step 4	Identify the species seeds for the current population according to the steps in Table I.
Step 5	For each species, run a basic DE. If a species has less than 10 members, randomly create new solutions in the species. If a child's fitness value is the same as that of its species seed, replace this child with a randomly generated new solution.
Step 6	Select the *NP* (population size) individuals from the combined population.
Step 7	Stop if the stop criteria are met, otherwise go to step 3.

B. Neighborhood based SDE

The neighborhood based SDE was introduced in 2012 to improve the performance of the original SDE [1]. It incorporated the neighborhood mutation concept into SDE to restrict the mutation and selection within a Euclidean distance based neighborhood. The neighborhood based mutation is able to split the searching resources and maintain multiple stable niches. The steps of NSDE algorithm is listed in Table III.

TABLE III. THE NSDE ALGORITHM

Step 1	Randomly generate *NP* number of initial trial solutions.
Step 2	Sort all solutions in descending order according to their fitness values.
Step 3	While the sorted population is not empty Find the species seed which is the best (objective value) unprocessed individual. Determine the most similar (in Euclidean distance) *m* individuals of the species seed and set them as one species. Remove the processed members for the current populations. Endwhile
Step 4	For each species, execute a global DE variant: If the child's fitness is the same as its species seed, replace this child with a randomly generated new individual.
Step 5	Select the *NP* (population size) individuals from the combined population.
Step 6	Stop if the stop criteria are met, otherwise go to step 3.

IV. EXPERIMENTAL PREPARATIONS

A. Experimental Setting

Matlab 2013 is selected as the programming tool. The computer configurations are Intel® Core™ i5 CPU and 4 GB of memory. The operating system is Microsoft Windows 7. The neighborhood size is selected as 5, i.e., m=5. The population sizes are chosen as 100, 250, 500 for 3 kinds of dimensions. The DE parameters used are list as below:

$$F=0.9,\ CR=0.1$$

Note that the number of function evaluations is defined by using the following formula $2000*D*\sqrt{q}$, where q is the number of optima number and D is the dimension of the problem. For example, for function 1, q=16, $MaxFES$=2000*5*4=20000.

B. Performance measure

Average number of optima found is used as the assessment criterion using the given level of accuracy. A level of accuracy, typically $0<\varepsilon<1$, is a value indicating how close the computed solutions to the known global peaks are. If the difference from a computed solution to a known global optimum is below ε , then the peak is considered to have been found. In this paper, the level of accuracy is equal to 0.1.

V. EXPERIMENTAL RESULTS

The simulation results of NSDE and SDE are presented in Table IV and V respectively. The comparison of these two algorithms on the mean value are listed in Table VI. As we can be seen from these results, NSDE outperform SDE on all the problems except those problems that both algorithms are not able to find any high quality solution.

In order to determine the statistical significance of the advantage of NSDE over SDE, t-test is applied on the average number of optima found and the results are shown in the last column of Tables VI. The numerical values -1, 0, 1 represent that the NSDE is statistically inferior to, equal to and superior to the SDE algorithm. As can be seen, there is no -1 in the table which means NSDE perform either better or equal to SDE for all the test functions. The superior performance of NSDE is because of the neighborhood mutation technique. Neighborhood mutation technique is able to distribute the searching resources over the search space. This will lead to the forming and maintaining of multiple niches which is suitable for multi-modal optimization problems. However, this operation will slow down the single global peak optimization as the searching resource are diversified.

TABLE IV. RESULTS OF NSDE

Func.	Dimension	Best	Worst	Mean	Std
1	5	0	3	1.60	0.9129
	10	0	0	0	0
	20	0	0	0	0
2	2	3	4	3.64	0.4899
	5	2	6	4	1
	8	15	17	16.2	1.0954
3	2	21	25	23.12	1.1662
	3	27	43	34.92	4.7864
	4	24	41	31.2	7.0852
4	5	0	3	0.40	0.7638
	10	0	0	0	0
	20	0	0	0	0
5	2	20	25	22.72	1.2754
	3	0	0	0	0
	4	0	0	0	0
6	4	7	12	9.28	1.2754
	6	10	20	13.84	2.6721
	8	14	22	16.60	3.1305
7	6	0	5	2.4	1.1902
	10	0	2	0.16	0.4726
	16	0	0	0	0
8	10	15	25	18.8	2.2174
	20	40	63	49.32	5.5882
	30	25	38	31	5.1479

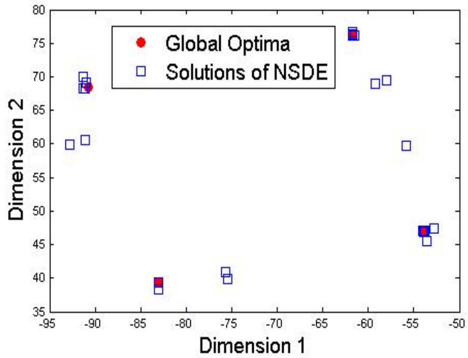
TABLE V. RESULTS OF SDE

Func.	Dimension	Best	Worst	Mean	Std
1	5	0	1	0.4	0.5
	10	0	0	0	0
	20	0	0	0	0
2	2	0	2	1.16	0.5538
	5	0	1	0.2	0.4472
	8	0	1	0.4	0.5477
3	2	0	5	2.24	1.20
	3	0	9	2.6	3.9749
	4	0	6	1.2	2.6833
4	5	0	0	0	0
	10	0	0	0	0
	20	0	0	0	0
5	2	0	0	0	0
	3	0	0	0	0
	4	0	0	0	0
6	4	0	2	1	0.5
	6	0	2	1.2	0.8367
	8	0	4	1.8	1.4832
7	6	0	1	0.04	0.20
	10	0	0	0	0
	16	0	0	0	0
8	10	1	6	3.12	1.2014
	20	2	7	4.8	1.9235
	30	2	4	2.8	0.8367

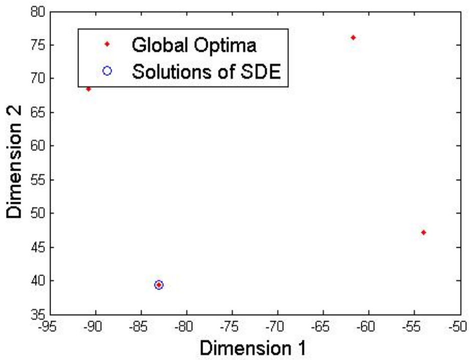
TABLE VI. THE COMPARISON OF NSDE AND SDE ALGORITHMS ON THE MEAN VALUES

Func.	Dimension	NSDE	SDE	*t*-test
1	5	1.60	0.4	1
	10	0	0	0
	20	0	0	0
2	2	3.64	1.16	1
	5	4	0.2	1
	8	16.2	0.4	1
3	2	23.12	2.24	1
	3	34.92	2.6	1
	4	31.2	1.2	1
4	5	0.40	0	1
	10	0	0	0
	20	0	0	0
5	2	22.72	0	1
	3	0	0	0
	4	0	0	0
6	4	9.28	1	1
	6	13.84	1.2	1
	8	16.60	1.8	1
7	6	2.4	0.04	1
	10	0.16	0	1
	16	0	0	0
8	10	18.8	3.12	1
	20	49.32	4.8	1
	30	31	2.8	1
No. Of ones				17

To further illustrate the superior performance of NSDE over SDE. The final solutions (parametric space) of NSDE and SDE are plotted for problem 2 and 3 (only 2D). The results are presented in Fig. 2 and 3. As can be see from these figures, NSDE is able to multiple niches through the whole search space. For SDE, it is more likely to converge to one or a few points which is bad for multi-modal optimization.

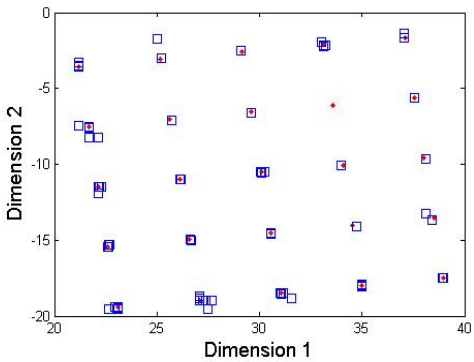


(a) NSDE

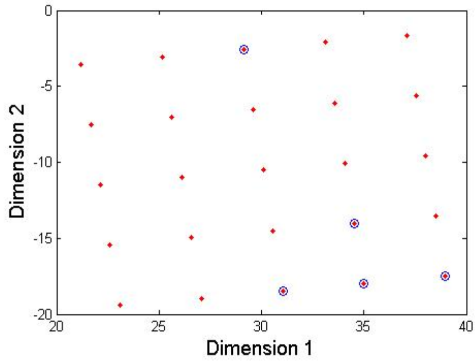


(b) SDE

Fig. 2 The final solutions of NSDE and SDE on problem 2 (2D)



(a) NSDE



(b) SDE

Fig. 3 The final solutions of NSDE and SDE on problem 3 (2D)

VI. CONCLUSION

DE is a effective algorithm in solving single global optimization problems. However, it need to be incorporated in certain niching technique in order to successfully solve multi-modal optimization problems. Neighborhood based species DE is a popular DE based niching algorithm. This work demonstrated the effectiveness of using NSDE to solve multi-modal problems. NSDE and the original SDE are evaluated by using the CEC2015 competition problems. The results reveal that NSDE is able to generate superior and more stable solutions.

References

[1] B. Y. Qu, P. N. Suganthan and J. J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, *IEEE Transaction on Evolutionary Computations*, vol. 16(5), pp. 601-614, 2012.

[2] G. Singh and K. Deb, "Comparison of multimodal optimization algorithms based on evolutionary algorithms," in Proc. Genet. Evol. Comput.Conf., 2006, pp. 1305–1312.

[3] B. Y. Qu, J. J. Liang and P. N. Suganthan, "Niching Particle Swarm Optimization with Local Search for Multi-modal Optimization," *Information Sciences*, 197: 131-143, 2012.

[4] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in Proc. IEEE Congr. Evol. Comput., Jun. 2004, pp.1382–1389.

[5] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in Proc. 2nd Int. Conf. Genet.Algorithms, 1987, pp. 41–49.

[6] G. R. Harik, "Finding multimodal solutions using restricted tournament selection," in Proc. 6th Int. Conf. Genet. Algorithms, 1995, pp. 24–31.

[7] A. Petrowski, "A clearing procedure as a niching method for genetic algorithms," in Proc. 3rd IEEE Congr. Evol. Comput., May 1996, pp.798–803.

[8] X. Li, “Efficient differential evolution using speciation for multimodal function optimization,” *in Proceedings of the conference on genetic and evolutionary computation*. Washington DC, USA, pp. 873-880, 2005.

[9] R. Storn and K. V. Price, "Differential evolution-Asimple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization, vol. 11, pp. 341-359, 1995.*

[10] B. Y. Qu, P. N. Suganthan, “Multi-Objective Differential Evolution with Diversity Enhancement”, *Journal of Zhejiang University-SCIENCE C*, Vol. 11, No. 7, pp. 538-543, 2010.

[11] S. Das and P. N. Suganthan, “Differential evolution: A survey of the state-of-the-art,” *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31,Feb. 2011.

[12] B. Y. Qu and P. N. Suganthan, “Modified species-based differential evolution with self-adaptive radius for multi-modal optimization,” *ICCP*, China, 2010.