

# Multimodal Optimization using Crowding Differential Evolution with Spatially Neighbors Best Search

Dingcai Shen

State Key Laboratory of Software Engineering, Computer School, Wuhan University, Wuhan, China  
School of Computer and Information Science, Hubei Engineering University, Xiaogan, China  
Email: d.c.shen@163.com

Yuanxiang Li

State Key Laboratory of Software Engineering, Computer School, Wuhan University, Wuhan, China

**Abstract**—Many real practical applications are often needed to find more than one optimum solution. Existing Evolutionary Algorithm (EAs) are originally designed to search the unique global value of the objective function. The present work proposed an improved niching based scheme named spatially neighbors best search technique combine with crowding-based differential evolution (SnbDE) for multimodal optimization problems. Differential evolution (DE) is known for its simple implementation and efficient for global optimization. Numerous DE-variants have been exploited to resolve diverse optimization problems. The proposed method adopts DE with DE/best/1/bin scheme. The best individual in the adopted scheme is searched around the considered individual to control the balance of exploitation and exploration. The results of the empirical comparison provide distinct evidence that SnbDE outperform the canonical crowding-based differential evolution. SnbDE has been shown to be efficient and effective in locating and maintaining multiple optima of selected benchmark functions for multimodal optimization problems.

**Index Terms**—Differential evolution, Multimodal optimization, Niching, Crowding

## I. INTRODUCTION

Most existing evolutionary algorithms (EAs) are aiming to find a single global optimum of an optimization problem. A large number of works have been done to improve the capability of the EAs to effectively and efficiently find the global value of the optimization problems. Among the commonly used evolutionary algorithms, differential evolution (DE) [1] is one of the most popular evolutionary algorithm for its simple and efficient. As with other EAs, DE is also a population-based stochastic optimization technique for solving continues optimization problems, especially for real-parameter optimization [2]. And the main purpose, as other techniques, is to find the single global value. However, in real-world applications, there often exist multiple optima (global or local), and it is often requested to locate either all or most of these solutions for more

options. For example, in the field of mechanical design we may choose the less fit solutions as our final choice due to physical or spatial restrictions that the optimal solutions may be hard to fabricate, or for the easiness of maintenance, or the reliability and so on.

Multimodal problems are usually thought as a difficult work to solve by canonical EAs because of the existence of multiple global or local optima. Numerous techniques have been applied to EAs to enable them to find and maintain multiple optima among the whole search space. These techniques can be classified into two major categories [3][4]: (i) iterative methods [5][6], which adopting the same optimization algorithm repeatedly to acquire multiple optima of a multimodal optimization problems; (ii) parallel subpopulation models (explicit or implicit), which achieving multiple solutions for a multimodal optimization problems by dividing a population into several groups, species or niches that evolve in parallel, such as Multinational GA [7], Species conservation technique [3][8][9], Crowding [10][11][12], AFMDE [13], fitness sharing [14], and so on. The canonical crowding and fitness sharing are widely used techniques but at the risk of losing niches during the evolution.

In this study, a new multimodal optimization method employs a spatially near best search strategy using crowding-based differential evolution (SnbDE) to find multiple optimal solutions simultaneously. The advantage of SnbDE is weakening the effect of the parameter named crowding factor (CF), which inappropriate setting may cause the replacement error. Numerical experiments on a set of widely used multimodal benchmark functions show that the SnbDE is able to find effective solutions to all tested functions.

The rest of this paper is outlined as follows. The next section presents relevant work, including crowding scheme used for multimodal optimization problems, and a brief review of differential evolution and its commonly employed variants. The proposed SnbDE is described in detail in section 3. In section 4, the proposed algorithm is compared with canonical crowding based DE [12] on

commonly used multimodal optimization benchmark functions. Finally, section 5 presents the conclusions and some future work.

## II. BACKGROUND AND RELATED WORK

This section presents a brief review of differential evolution and its extension to use crowding scheme for solving multimodal optimization problems.

### A. Differential Evolution

Differential evolution (DE) was first proposed by Storn and Price [1] in 1995. The DE gained significant popularity since its original definition because of its simplicity and easiness of implementation and efficiency. Like other popular used EAs, DE is a population-based algorithm designed to solve optimization problems over continuous search space.

Without loss of generality, we refer to the maximization problem, both global and local, of an objective function  $f(x)$  throughout this paper, where  $x$  is a vector of  $D$ -dimensional variables in feasible solution space, which can be denoted as  $\Omega = \prod_{i=1}^D [L_i, U_i]$ . At the initialization step, a population  $x_i^0 = (x_{i,1}^0, x_{i,2}^0, \dots, x_{i,D}^0)$ ,  $i = 1, 2, \dots, NP$ , is randomly generated from the feasible solution space, where  $NP$  is the population size.

Following initialization, the mutation operation is conducted with the creation of a mutant vector  $v_i^t = (x_{i,1}^t, x_{i,2}^t, \dots, x_{i,D}^t)$  for each individual  $x_i^t$  in current population. The five widely used mutation strategy can be described as follows.

1) DE/rand/1:

$$v_i^t = x_{r1}^t + F \cdot (x_{r2}^t - x_{r3}^t) \quad (1)$$

2) DE/best/1:

$$v_i^t = x_{best}^t + F \cdot (x_{r1}^t - x_{r2}^t) \quad (2)$$

3) DE/current-to-best/1:

$$v_i^t = x_i^t + F \cdot (x_{best}^t - x_i^t) + F \cdot (x_{r1}^t - x_{r2}^t) \quad (3)$$

4) DE/best/2:

$$v_i^t = x_{best}^t + F \cdot (x_{r1}^t - x_{r2}^t) + F \cdot (x_{r3}^t - x_{r4}^t) \quad (4)$$

5) DE/rand/2:

$$v_i^t = x_{r1}^t + F \cdot (x_{r2}^t - x_{r3}^t) + F \cdot (x_{r4}^t - x_{r5}^t) \quad (5)$$

where  $x_{best}^t$  denote the fittest individuals of the generation  $t$ ,  $r1, r2, r3, r4, r5$  are integers randomly drawn from the discrete set  $\{1, 2, \dots, NP\}$  and not equal to the index of the considered individual, and  $F > 0$  is the mutation factor used to scale differential vectors.

The crossover operation is applied subsequently on each component  $j (j = 1, 2, \dots, D)$  of the mutant

individual  $v_{i,j}^t$ . The process of the crossover is depicted as follows:

$$u_{i,j}^t = \begin{cases} v_{i,j}^t, & \text{if } rand_j(0,1) \leq Cr \text{ or } j = j_{rnd} \\ x_{i,j}^t, & \text{otherwise.} \end{cases} \quad (6)$$

where  $rand_j(0,1)$  is the uniform random variable drawn from the interval  $[0,1]$ ,  $j_{rnd}$  is a random integer number in  $\{1, 2, \dots, NP\}$ , and  $Cr \in [0,1]$  is the crossover constant namely crossover rate or recombination factor.

The selection operator is performed at the last step of DE to sustain the most promising trial individuals into the next generation. The selection operator can be defined as

$$x_i^{t+1} = \begin{cases} u_i^t, & \text{if } f(u_i^t) > f(x_i^t) \\ x_i^t, & \text{otherwise.} \end{cases} \quad (7)$$

The DE algorithm performs repeated applications of the above process until a specified number of iterations have been reached, or the number of maximum evaluations has been exceeded. The pseudo-code of canonical differential evolution algorithm is shown in Algorithm 1.

---

#### Algorithm 1 The algorithm of canonical Differential Evolution (DE/best/1)

---

```

1:  $iter \leftarrow 0$ 
2: Initialize population  $P$  with  $NP$  randomly generated individuals.
3: Evaluate individual  $P_i, i = 1, 2, \dots, NP$ .
4: while (termination criteria are not satisfied) do
5:   Find the best fit individual  $P_{best}$  of current population
6:   for  $i = 1$  to  $NP$  do
7:     Select  $r1, r2 \in \{1, 2, \dots, NP\}$  and  $r1 \neq r2 \neq i$ 
8:     Copy individual  $P_i$  to  $S_i$ 
9:      $n = rand(1, D)$ 
10:    for  $j = 1$  to  $D$  do
11:      if  $rand(0,1) < Cr$  or  $j = D$ 
12:         $S_i[n] = P_{best}[n] + F * (P_{r1}[n] - P_{r2}[n])$ 
13:      end if
14:       $n \leftarrow (n + 1) \bmod D$ 
15:    end for
16:    Evaluate individual  $S_i$ 
17:    if  $f(S_i) < f(P_i)$ 
18:       $S_i \leftarrow P_i$ 
19:    end if
20:    if  $f(S_i) > f(P_{best})$ 
21:       $P_{best} \leftarrow S_i$ 
22:    end if
23:  end for
24:   $iter \leftarrow iter + 1$ 
25: end while

```

---

### B. Crowding

De Jong [15] first presented his conception of crowding to deal with multimodal optimization problems. Crowding is inspired by the natural phenomenon of the competition amongst similar members for limited resources. The proposal of crowding technique is aimed to preserve the diversity of the population. In order to achieve this goal, a subset of the population is randomly chosen to be replaced by the most similar individuals of the offspring if it is fitter. The determination of similarity is judged by the distance metric either in genotypic or phenotypic search space. In real-valued coding optimization problems, the Euclidean distance between two points is most widely used. The number of individuals selected to be considered for replacement is decided by the parameter named crowding factor (CF). For low CF values, a common problem named replacement error may emerge which is caused by the offspring replaces another individual that may not be similar to the offspring [12][16].

To reduce the replacement error problem caused by canonical crowding with low CF values, a number of works have been done to overcome this problem. Such as Deterministic crowding proposed by [17], Probabilistic crowding [18], or just set CF equal to the number of individuals in the population as crowding DE (CDE) [12] did. The algorithm of CDE can be depicted as Algorithm 2.

---

#### Algorithm 2 The algorithm of crowding DE

---

```

1:  $t \leftarrow 0$ 
2: Initialize population  $P$  with  $NP$  randomly generate individual
3: Evaluate  $P_i, i = 1, 2, \dots, NP$ 
4: while ( not termination condition) do
5:   Use the conventional DE to generate offspring  $S$ 
6:   for all Individual  $i$  in  $S$  do
7:     Calculate the Euclidean distance values of the individual  $S_i$  to the individuals in population  $P$ .
8:     Find the nearest individual  $P_j$  in population  $P$  to individual  $S_i$ .
9:     if  $f(S_i) > f(P_j)$ 
10:      Replace individual  $P_j$  with individual  $S_i$ .
11:   end if
12: end for
13:  $t \leftarrow t + 1$ 
14: end while

```

---

### III. SNBDE

As previously mentioned, DE, similar to other EAs, has shown highly efficient for global optimization problems. However, DE is incapable to locate multiple global optima or detect other desirable suboptima simultaneously. The CDE which extend DE with the crowding scheme [12] shows good performance in

locating peaks and maintaining found peaks throughout the run. CDE produce many niches around the found optima (global or local). Ideally, all the individuals are gathered around the global optima when the algorithm terminated. However, since the polytropic size of basin of attraction of various multimodal optimization problems, some suboptima may be vanished during the process of evolution.

DE, as a population-based stochastic search algorithms, should consider two crucial contradictory aspects, *exploration* and *exploitation* [19], in order to guarantee acceptable success on a given task. For multimodal problems, the ability of exploration impulse the algorithm to explore every area of the feasible search space so as to locate the global optima, while the ability of exploitation guarantee converging to the nearby optima thus maintaining of found optima. The DE variant known as DE/best/1 adopt the best fit vector of the current population to generate donor vectors. The scheme enhances exploitation ability since all the vectors are attracted by the single global vector. At the same time, this scheme weakens the exploration abilities, and may get trapped in some local optima.

In this study, we propose a new algorithm named spatially neighbors best search crowding differential evolution (SnbDE). Different from the ring topology of neighborhood, which use indices based neighborhood. Let's consider a DE population  $P = [X_1, X_2, \dots, X_{NP}]$ , where each  $X_i (i = 1, 2, \dots, NP)$  is a D-dimensional vector in the feasible solution space. The vector indices are randomly generated, say as their generating order at population initialization step, and are kept constant throughout the run. A vector  $X_i$ 's  $r (0 < r < (NP - 1) / 2)$  neighborhoods are  $X_{i-r}, \dots, X_i, \dots, X_{i+r}$ . Note that topological neighborhoods may belong to a different peak in the fitness landscape because they were chosen randomly (in terms of fitness landscape). In order to overcome this problem, the SnbDE algorithm using a spatial neighborhood based on Euclidean distance around the considered individual.

The SnbDE is based on the CDE but differently in the generating of offspring. In SnbDE, we use the /DE/best/1/ scheme, where the "best" represent the best fit individual around the current optima. The crowding with nearest neighbors replacement proposed by Li et al. [11] was similar to the crowding scheme with spatially neighbors best search strategy proposed in this study. In [11], the lowest fitness individual in the selected  $q$ -nearest subset based on the Euclidean distance will be replaced by the offspring if it less fit. In our proposed algorithm, in order to enhance the exploitative ability, in the meanwhile, to keep the explorative ability of crowding DE, similar process as proposed in [11] is adopted. For each individual  $x_i, (i = 1, 2, \dots, NP)$  in population  $P$ , the individuals in population  $P$  are sorted by ascending order based on the Euclidean distance  $d(x_i, x_j) (i, j = 1, 2, \dots, NP; i \neq j)$ . Thus we get

the reordered set denoted as  $P_s = \{x_1, x_2, \dots, x_{NP-1}\}$ . The local best individual  $x_{lbest}$  is searched in previously  $r(r \leq NP)$  individuals in  $P_s$ . Use  $x_{lbest}$  as the best individual in /DE/best/1 scheme to generate offspring, then the canonical crowding DE is conducted to select individuals thus produce the new population. This procedure is described as Algorithm 3.

---

**Algorithm 3 The illustration of spatially near best search crowding DE**

---

```

1: Initialize population  $P$  with  $NP$  random individuals
2: Evaluate individuals in  $P$ 
3:  $gen \leftarrow 1$ 
4: while (not termination condition) do
5:   for  $i = 1$  to  $NP$ 
6:     for  $j = 1$  to  $NP$ 
7:       Calculate Euclidean distance  $d(x_i, x_j), i \neq j$ 
8:     end for
9:     Sort  $P_i$ 's neighbors set in ascending order,
        $P_s = \{x_1, x_2, \dots, x_{NP-1}\}$ 
10:    Choose the previously  $r$  individuals,
        $Pr = \{x_1, x_2, \dots, x_r\}$ 
11:    Find the best individual  $x_{lbest} \in Pr$ 
12:    Using  $x_{lbest}$  as the local best individual in
       /DE/best/1 scheme to generate offspring  $S_i$ 
13:  end for
14:  Using crowding procedure described in
       algorithm 2 to generate new population
15:   $gen \leftarrow gen + 1$ 
16: end while

```

---

#### IV. EXPERIMENT RESULTS

This section presents results obtained by applying SnbDE to search multiple optima on a set of multimodal problems. Then, the performance of the proposed algorithm and standard crowding DE are tested on these functions. The effect of different control parameter values for SnbDE is also investigated.

##### A. Test Functions

The benchmark functions used in this study are widely used for comparing the performance of multimodal optimization problems. The number of optima (global or local) of the functions is ranging from 2 to 6, dimensionality is from 1 to 2, and the high dimensionality problem is our future research. These functions are described as follows.

###### 1) Deb's 1st function

Deb's 1st function is defined as:

$$F_1(x) = \sin^6(5\pi x) \quad (8)$$

where  $0 \leq x \leq 1$ . Deb's 1st function has 5 equally spaced global optima as showed in Fig. 1. All five global values are uniformly distributed at  $x = 0.1, 0.3, 0.5, 0.7, 0.9$  with a value 1.0.

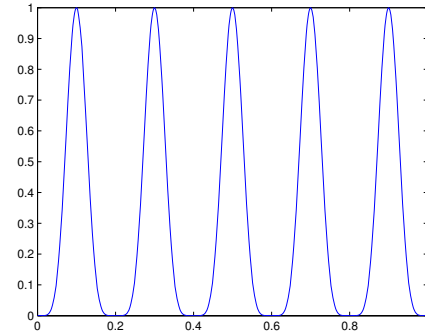


Figure 1. Deb's 1st function

###### 2) Deb's decreasing function

Deb's decreasing function can be described as:

$$F_2(x) = 2^{-2 \ln(2)((x-0.1)/0.9)^2} \sin^6(5\pi x) \quad (9)$$

Deb's decreasing function is similar to Deb's 1st function, with the values corresponding to the optima are 1.0, 0.934, 0.760, 0.540 and 0.334, respectively (see Fig. 2).

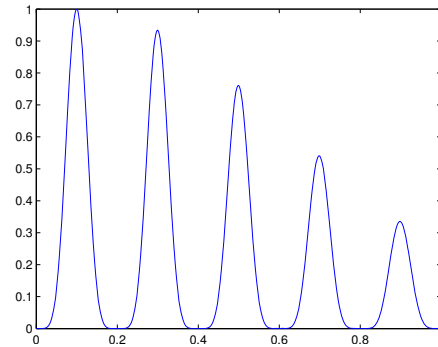


Figure 2. Deb's decreasing function

###### 3) Branin RCOS

$$F_3(x, y) = \left(y - \frac{5.1}{4\pi^2}x^2 + \frac{5}{\pi}x - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x) + 10 \quad (10)$$

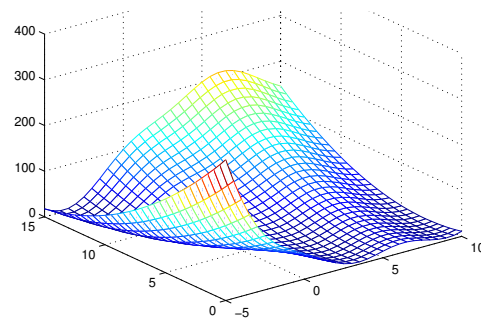


Figure 3. Branin RCOS

The Branin RCOS function is a two-dimensional function with 3 global optima, which optimum value is

0.397887 in the region  $-5 \leq x \leq 10$  and  $0 \leq y \leq 15$  (see Fig. 3).

#### 4) Ursem's F4

$$F_4(x, y) = 3 \sin(0.5\pi x + 0.5\pi) \frac{(2 - \sqrt{x^2 + y^2})}{4} \quad (11)$$

where  $-2 \leq x, y \leq 2$ . Ursem's F4 is a two-dimensional function with one global optimum and four local peaks. All the four local optima are distributed on the edge of the search space and the global one in the center (see Fig. 4).

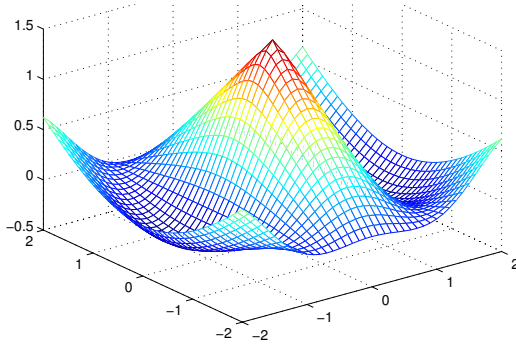


Figure 4. Ursem's F4

#### 5) Six-Hump Camel Back

$$F_5(x, y) = -((4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + \frac{4(y^2 - 1)}{y^2}) \quad (12)$$

where  $-1.9 \leq x \leq 1.9$  and  $-1.1 \leq y \leq 1.1$ . The six-hump camel back function has six optima (2 global and 4 local, see Fig. 5). This function is used by many multimodal optimization algorithms to inspect the effectiveness to locate all optima. Because among the four local optima, two of them are stable, the other two are unstable, and all optima are near from each other. The specific characteristic makes it hard to identify all the optima by most multimodal optimization algorithm.

#### B. Performance Measures

Several performance measures for multimodal optimization algorithms have been used in literature. The widely used performance measures are described briefly as follows:

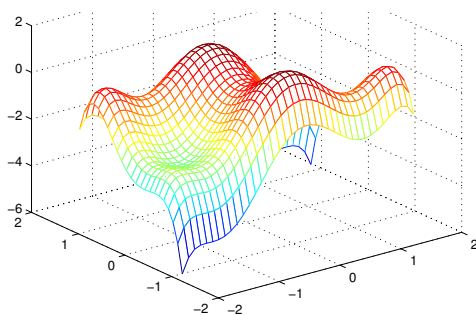


Figure 5. Six-hump camel back

- **Success rate (%)**: the percentage of successful runs in which all optima (global and local) are successfully found.
- **Peak ratio**: measure the ability of algorithms to locate the multiple optima, the sum of the optima identified by the technique divided by the sum of the actual optima in the search space. In this study, an optimum is considered to be detected if it is within a Euclidean distance of 0.01 for  $F_1, F_2$  and 0.05 for  $F_3, F_4, F_5$  from the real optimum.
- **NFEs**: The number of fitness function evaluations is recorded when the pre-assigned threshold is reached. The average and standard deviation of the number of fitness evaluations are used in this work.

#### C. Results and Analysis

Table I presents descriptive statistics related to the five benchmarks, i.e. the values for the mean of the average success rate, the mean and standard deviation of the fitness evaluation and the mean peak ratio. The specific parameter setting for each of the algorithms are described as follows. Population size  $NP$  was set to 30 for  $F_1, F_2$ , and 50 for  $F_3, F_4$  and 100 for  $F_5$ . The probability of crossover was fixed to  $Cr = 0.9$ , and the scaling factor was  $F = 0.5$ . The parameter  $r$  was set to 2, and an investigation of the effect of different  $r$  to the performance of SnbDE will be provided. The crowding factor for both SnbDE and CDE were set to equal to the population size ( $CF = NP$ ).

TABLE I.  
EXPERIMENTAL RESULTS(AVERAGED OVER 30 RUNS)

| Func. |       | Success Rate (%) | NFEs            | Peak Ratio |
|-------|-------|------------------|-----------------|------------|
| $F_1$ | CDE   | 100              | $590 \pm 163$   | 1.0        |
|       | SnbDE | 100              | $410 \pm 145$   | 1.0        |
| $F_2$ | CDE   | 100              | $680 \pm 230$   | 1.0        |
|       | SnbDE | 100              | $570 \pm 215$   | 1.0        |
| $F_3$ | CDE   | 100              | $7955 \pm 560$  | 1.0        |
|       | SnbDE | 100              | $5850 \pm 515$  | 1.0        |
| $F_4$ | CDE   | 100              | $1850 \pm 125$  | 1.0        |
|       | SnbDE | 100              | $1350 \pm 115$  | 1.0        |
| $F_5$ | CDE   | 68               | $8250 \pm 1520$ | 0.67       |
|       | SnbDE | 100              | $5210 \pm 1210$ | 1.0        |

We compare the results obtained using SnbDE with canonical CDE. Each algorithm is tested on each function for 30 runs. From the Table I we can see SnbDE is able to identify and maintain multiple optima in the search space. Our approach is comparatively successful to the canonical CDE in all tested functions. Especially for  $F_5$ , in all 30 runs, SnbDE successfully locate and maintain the six optima, while CDE can only find the 4 global optima in most runs.

As Fig. 6 indicates, because of the global selection of CDE, some found peaks may be attracted by the higher fitter individuals result in the missing of optima. From Fig. 6(a), we can see the initial population is uniformly distributed among the search space, after the algorithm

evolved 150 generation; most individuals gathered around the four global optima, the local individuals are attracted by the global optima. When the algorithm attained its 800 generations, all individuals are gathered around the four global optima. This status retained to the end of the evolution. With the same distribution of population, however, the SnbDE successfully identified and maintained all six optima.

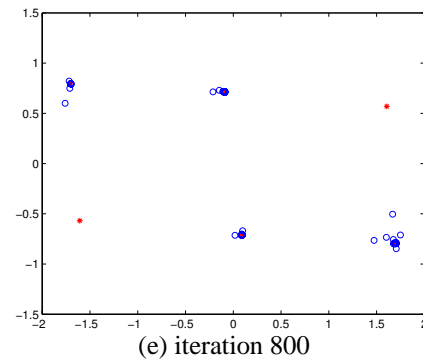
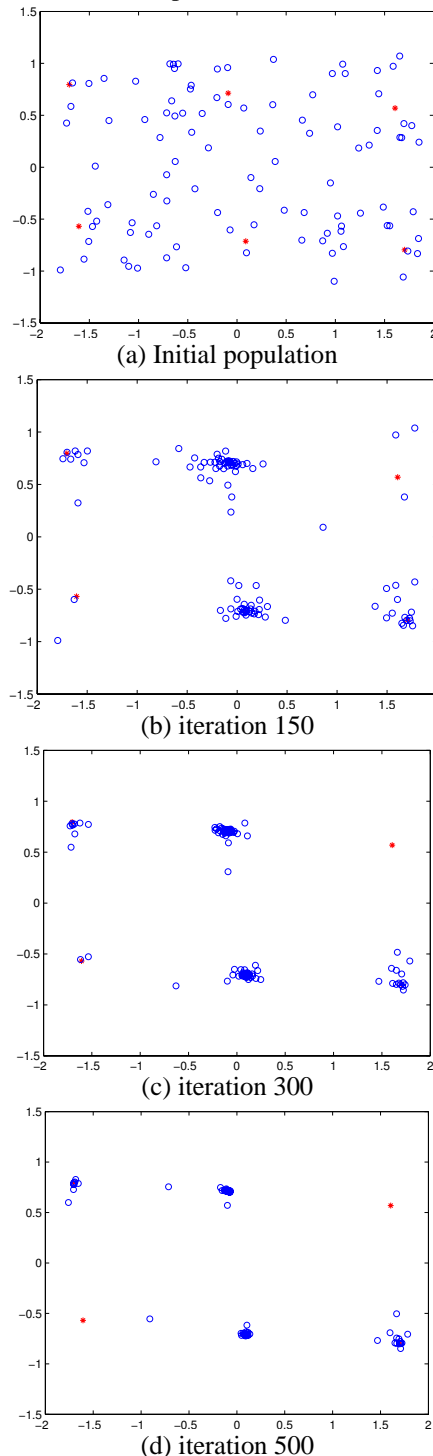


Figure 6. Individuals distribution of CDE on the  $F_5$  (with a population size of 100), asterisk represents the real optima(global or local)

Fig. 7 shows the distribution of individuals when the algorithm achieved the pre-specified threshold with the same initialization population.

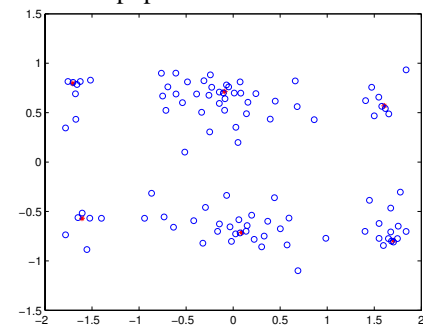


Figure 7. The distribution of SnbDE on  $F_5$  after 44 iterations (with a population size of 100), asterisk represents the real optima(global or local)

Fig. 8 illustrate the effect of population size on peak ratio (with  $r = 2$  for SnbDE) when the algorithms applied to  $F_5$ . From Fig. 8 we can see, with the increasing of population size, the peak ratio attained of both algorithms is increased. When the population size was set too small (e.g. less than or equal to twenty), the peak ratio achieved by CDE is higher than that of SnbDE, with the increasing of population size, the peak ratio achieved by SnbDE growing more rapidly than that of CDE.

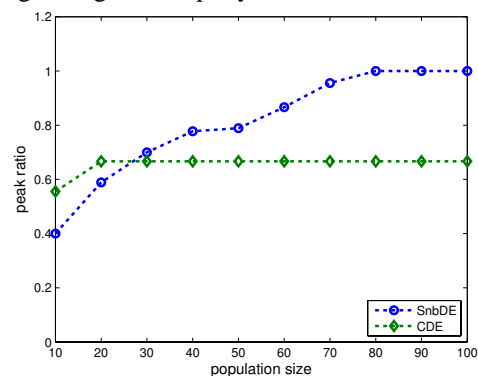


Figure 8. The effect of population size on peak ratio of the  $F_5$

When the population size reached 80, the peak ratio come to 1.0 for SnbDE, and about 0.67 for CDE, respectively. When the population size reached 100, the



peak ratio reached 1.0 for SnbDE, and CDE keep constant at about 0.67. In our experiments, we also notice that once a new optimum found, the SnbDE can maintain it throughout the run, while CDE often lose the found local optimum, and in most cases can only maintained four optima.

The effect of  $r$  to the peak ratio of algorithm SnbDE on  $F_5$  is depicted as Fig. 9. As can be seen, there was a significant effect for  $r$  value on peak ratio. With a lower  $r$  (e.g.  $r=2$ ), the SnbDE can locate and maintain all optima. With the increasing of  $r$ , the selection of best individual became the global selection result in the gathering around the global optima.

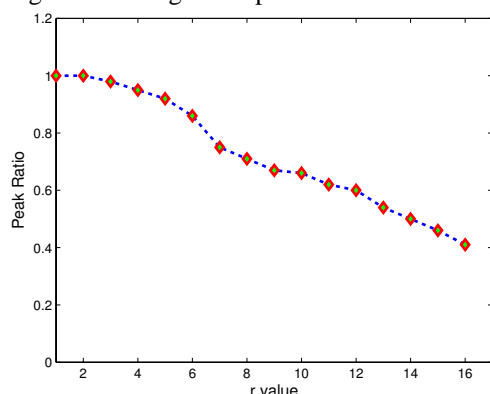


Figure 9. The effect of  $r$  on peak ratio of SnbDE on  $F_5$

## V. CONCLUSION

In this paper we present the SnbDE algorithm based on the idea of crowding DE with spatially nearest best search strategy. The performance of the SnbDE is compared against the canonical CDE on several commonly used multimodal benchmark functions. All experiments have demonstrated that the SnbDE is a competitive candidate for multimodal evolutionary algorithm. Further research will investigate whether SnbDE is effective in handling complex real-world multimodal optimization problems, including ones with higher dimensionality. The ability to track optima in a dynamic environment will also be an interesting topic for future research.

## ACKNOWLEDGMENT

This work is supported by the National Nature Science Foundation of China (Grant No.61070009, 61165004).

## REFERENCES

- [1] R. Storn and K. V. Price, "Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces," *International computer science institute*, vol. 46, no. 2. Institute of Company Secretaries of India, 24-Jun-1995.
- [2] U. K. Chakraborty, *Advances in differential evolution*. Springer, 2008.
- [3] J.-P. Li, X.-dong Li, and A. Wood, "Species based evolutionary algorithms for multimodal optimization: A brief review," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1-8.
- [4] J. Yao, N. Kharm, and P. Grogono, "Bi-Objective Multipopulation Genetic Algorithm for Multimodal Function Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 80-102, Feb. 2010.
- [5] J. E. Vitela and O. Castaños, "A sequential niching memetic algorithm for continuous multimodal function optimization," *Applied Mathematics and Computation*, Feb. 2012, doi:10.1016/j.amc.2011.05.051
- [6] D. Beasley, D. R. Bull, and R. R. Martin, "A Sequential Niche Technique for Multimodal Function Optimization," *Evolutionary Computation*, vol. 1, no. 2, pp. 101-125, Jun. 1993.
- [7] R. Ursem, "Multinational GAs: Multimodal optimization techniques in dynamic environments," in *Proc. of the Genetic and Evolutionary Computation Conference*, 2000, pp. 19-26.
- [8] C. Stoean, M. Preuss, R. Stoean, and D. Dumitrescu, "Multimodal Optimization by Means of a Topological Species Conservation Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 6, pp. 842-864, Dec. 2010.
- [9] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evolutionary computation*, vol. 10, no. 3, pp. 207-234, Jan. 2002.
- [10] M. Li, D. Lin, and J. Kou, "A hybrid niching PSO enhanced with recombination-replacement crowding strategy for multimodal function optimization," *Applied Soft Computing*, vol. 12, no. 3, pp. 975-987, Mar. 2012.
- [11] M. Li and J. Kou, "Crowding with nearest neighbors replacement for multiple species niching and building blocks preservation in binary multimodal functions optimization," *Journal of Heuristics*, vol. 14, no. 3, pp. 243-270, Sep. 2008.
- [12] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proceedings of the 2004 Congress on Evolutionary Computation*, 2004, vol. 2, pp. 1382-1389.
- [13] D. Shen, Y. Li, B. Wei, and X. Xia, "Adaptive Forking Multipopulation Differential Evolution Algorithm for Multimodal Optimization," *Journal of Convergence Information Technology*, vol. 7, no. 5, pp. 57-65, Mar. 2012.
- [14] A. Della Cioppa, C. De Stefano, and A. Marcelli, "Where Are the Niches? Dynamic Fitness Sharing," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 4, pp. 453-465, Aug. 2007.
- [15] K. D. Jong, "Analysis of the behavior of a class of genetic adaptive systems," University of Michigan, 1975.
- [16] S. Das, S. Maity, B.-Y. Qu, and P. N. Suganthan, "Real-parameter evolutionary multimodal optimization — A survey of the state-of-the-art," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 71-88, Jun. 2011.
- [17] S. Mahfoud, "Crowding and preselection revisited," in *Parallel Problem Solving From Nature*, 1992, pp. 27-36.
- [18] O. J. Mengshoel and D. E. Goldberg, "Probabilistic Crowding: Deterministic Crowding with Probabilistic Replacement," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999, pp. 409-416.
- [19] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential Evolution Using a Neighborhood-Based Mutation Operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526-553, Jun. 2009.