ISMSI 19 #015

Bat Algorithm with Dynamic Niche Radius for Multimodal Optimization

ABSTRACT

In this paper, we propose Bat Algorithm extending with Dynamic Niche Radius (DNRBA) which enables to solutions locate multiple local and global optima for solving multimodal optimization. This proposed algorithm is designed Bat Algorithm (BA) dealing with the exploration and the exploitation search with Niche Radius which is calculated by the fitness landscape and the number of local and global optima to avoid converging solutions on the same optimum. Although Niche Radius is an effective niching method for locating solutions at the peaks in the fitness landscape, it has not applicable for uneven multimodal functions and easily fail to keep multiple optima. To overcome this problem, we introduce dynamic niche sharing scheme which is able to adjust the distance of niche radius in the search process dynamically for BA. In the experiment, we employ several multimodal functions and compare DNRBA with conventional BA to evaluate the performance of DNRBA.

CCS CONCEPTS

• **Mathematics of computing → Bio-inspired optimization**;

KEYWORDS

Bat Algorithm, Multimodal optimization, Swarm Intelligence, Niching method

ACM Reference Format:

Takuya Iwase, Ryo Takano, Fumito Uwano, Hiroyuki Sato, and Keiki Takadama. 2019. Bat Algorithm with Dynamic Niche Radius for Multimodal Optimization. In *Proceedings of 3rd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence (ISMSI'18)*. ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

There are many studies on evolutionary algorithms (EAs) solving the real-world optimization problems which are mostly multimodal and complex optimization problems. These problems has not only single global optimum but also many local optima, hence EAs are required to find the both multiple optima which might be changed their own location as the environment changes. However, most of EAs such as Genetic Algorithm (GA) [5], Particle Swarm Optimization (PSO) [8] and Differential Evolution(DE) [9] designed to converge toward a single global optimum for the static environment, is difficult to find these optima.

To tackle multimodal optimization problems, various niching methods have proposed. Thomsen proposed the DE extends with a crowding scheme (CDE) [10] to replace the high-quality solution by the most similar candidate solutions. Li proposed the DE with Speciation (SDE) [4] to keep a solution away from the nearest neighbor solution when the distance of both these solutions is less than the threshold. However, these niching methods are not still enough to find multiple local optima because both of them do not consider searching globally though they consider the solution movement according to the euclidean distance between the nearest neighbor solutions. For solving this problem, this paper focuses on Bat Algorithm (BA) that has the characteristic of echolocation which can predict the distance between bats and the target (*i.e., object/food source*). This algorithm enables bats to estimate the distance between their location and the target even in the situations such as the target surrounded by obstacles and the absence of light. Bats can adjust their velocity which is controlled by their loudness, pulse emission rate and frequency, toward the target. While the iteration search step continues until bats reaches the target in the evolutionary process, bats will stop searching the target within its perceptible distance. This research employs BA which copes with exploitation and exploration search, extending with Niche Radius for multimodal optimization. Niche Radius is the threshold distance calculated by the fitness landscape and the number of its peaks.

This paper is organized as follows. After this section, the mechanism of BA and the proposed algorithm NRBA are explained in Sections 2 and 3. Section 4 describes the multimodal functions as

DNRBA??

the test-bed problem in the experiment. Section 5 shows the results while Section 6 discusses them. Finally, this paper concludes in Section 7.

2 BAT ALGORITHM

As mentioned in Section 1, BA is a metaheuristic algorithm based on the bat behavior with echolocation which are the loudness and the pulse emission rate of the reflect wave controlling the balance between the exploitation and the exploration search. As a bat approaches to a better solution than its current solution, BA turn the loudness A_i up and the pulse rate r_i down. The bats behavior is updated by the following three solution search phases: (i) the bat i flies to the target (*i.e.*, the bat which finds the best solution) with the flight speed controlled by frequency f_i; (ii) the bat i flies around the target as a local search; and (iii) the bat i flies randomly in the search space as a global search.

First, in the exploitation phase (i), all bats change their locations x_i with their velocities v_i toward the global best solution. For this calculation, the frequency f_i, velocity v_i, and location x_i, of the bat i are calculated as follows:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{1}$$

$$v_i^{t+1} = v_i^t + (x_* - x_i^t) * f_i \tag{2}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{3}$$

In detail, the new solution x_i is updated by adding the new the velocity v_i which is derived from the previous velocity v_i^t, the distance between the global best location and the previous location $x_* - x_i^t$, and frequency f_i which range is $[f_{min}, f_{max}]$ where $f_{min} = 0$ and $f_{max} = 1$. β is uniform random value from 0 to 1. Next, in the local search phase (ii), the new solution x_{loc} is generated around the global best solution as follows:

$$x_{loc} = x_* + \epsilon A^t \tag{4}$$

where ϵ is uniform random value between [0, 1]. In Eq.(6), A^t is the averaged loudness of all bats. Finally, in the global search phase (iii), x_{rnd} is generated randomly in search space as follows:

$$x_{rnd} = x_{lb} + (x_{ub} - x_{lb}) * rand(1, D) \tag{5}$$

where x_{ub} and x_{lb} describe the upper and lower bounds of the search space, and $rand(1, D)$ is the D dimensional uniform random value between [0, 1].

When a bat finds the better solution than the current one, the loudness A_i and pulse emission rate r_i are updated as follows:

$$A_i^{t+1} = \alpha A_i^t \tag{6}$$

$$r_i^{t+1} = r_i^0[1 - exp(-\gamma t)] \tag{7}$$

Note that the loudness A_i^0 is initialized as $A_i^0 = 1$ and the pulse rate is initialized as a uniform random value r^0 between [0, 1] or a number closed around zero. The parameters α and γ are the symbolized damping coefficient. The pseudo code of BA is given in the Algorithm 1 and its brief summary is described below.

- STEP1: Population initialization of bats (line 1 to 3)
  The population of bats $x_i (i = 1, 2, ..., N)$, the loudness A_i^0, the pulse rate r_i^0 are initialized as the initial values. The frequency f_i is initialized by Eq.(1).
- STEP2: New solution updates (line 6)
  The new solutions x_i is calculated by Eqs. (2)(3).
- STEP3: New solution generation around global best solution x_* (line 7 to 9)
  A new solution x_{loc} is generated around x_* by Eq. (4) when the pulse emission rate r_i is lower than a random value.
- STEP4: Random new solution generation (line 10)
  A new solution x_{rnd} is generated randomly by Eq. (5).
- STEP5: Solutions update(line 11 to 14)
  When $rand < A_i$, the best solution is selected from x_i, x_{loc}, and x_{rnd} by Eqs.(6)(7)
- STEP6: Return to STEP2

Algorithm 1 Bat Algorithm

Require: Objective Function $F(x)$
1: Initialize Population $x_i (i = 1, 2, ..., N)$ and v_i
2: Define frequency f_i at location x_i [eq.(1)]
3: Initialize pulse rates r_i, and loudness A_i
4: **while** ($t <$ Max number of iterations) **do**
5:   **for** i=1 to N **do**
6:     Generate a new solution x_i and velocity v_i [eqs.(2),(3)]
7:     **if** ($rand > r_i$) **then**
8:       Generate a new solution x_{loc} around a global best solution x_i [eq.(4)]
9:     **end if**
10:     Generate a new solution x_{rnd} randomly
11:     **if** ($rand < A_i \& \min(F(x_i), F(x_{loc}), F(x_{rnd}) < F(x_{i*}))$ **then**
12:       Accept the new solution, and update pulse rate r_i & loudness A_i [eqs. (6)(7)]
13:     **end if**
14:     Evaluate all bats and select a best solution x_* in the current solutions
15:   **end for**
16: **end while**

3 SHARING SCHEME

The first sharing mechanism was proposed by Holland [3] to spread solutions widely in the multimodal optimization. The concept of sharing scheme is to reduce the fitness value of similar individuals and classify these individuals within the population.

3.1 Niche Radius

The *niche radius* is the distance calculated by the upper and lower bounds of the search space. The equation is defined by:

$$dist = \frac{1}{2}\sqrt{(x_{ub} - x_{lb})^2} \tag{8}$$

$$\sigma = \frac{dist}{\sqrt[D]{q}}, \tag{9}$$

where x_{ub}, x_{lb} are the upper and lower bounds of the D-th dimensional search space, and q is the number of peaks of the fitness landscape.

how known?

3.2 Fitness Sharing

Fitness Sharing is derived from the concept of *crowding scheme* [2] replacing a new individual by ~~the~~ nearby similar individual in population. The most widely used ~~the~~ *sharing function* is given as follows:

$$sh(d_{ij}) = \begin{cases} 1 - (\frac{d_{ij}}{\sigma})^{\alpha} & (\text{if } d_{ij} < \sigma) \\ 0 & (\text{otherwise}) \end{cases} \tag{10}$$

where d_{ij} is the distance between individuals i, j, and σ is the niche radius defined above in eq.(9) as the threshold. α is the coefficient parameter, basically set to 1. By the *sharing function*, the *niche count* which represents the density of nearby similar individuals ~~*sharing function*~~, is defined by:

$$m_i = \sum_{j=1}^{N} sh(d_{ij}) \tag{11}$$

Subsequently, *The shared fitness* ϕ_i is given by:

$$\phi_i = \frac{F_i}{m_i} \tag{12}$$

where F_i is the raw fitness value of the individual and m_i is the niche count. *the shared fitness* indicates the fitness of the individual considering the density of similar individuals ~~nearby itself~~.

3.3 Dynamic Niche Sharing

In order to cut off the redundancy of the *shared fitness*, *dynamic niche sharing* is proposed by Miller in [6]. This scheme enables to identify the q peaks of the fitness landscape and classify all individuals into several groups in the same domain with the radius dynamically.

$$m_i^{dyn} = \begin{cases} n_j & (\text{if individual } i \text{ is within dynamic niche j}) \\ m_i & (\text{otherwise}) \end{cases} \tag{13}$$

where n_j is the $j-th$ niche radius and m_i is the *niche count* defined in eq.(11) as mentioned above. The *shared fitness* is calculated as follows:

$$\phi_i^{dyn} = \frac{F_i}{m_i^{dyn}} \tag{14}$$

4 PROPOSED ALGORITHM

4.1 Bat Algorithm with Dynamic Niche Radius

In our algorithm, we provide a new *dynamic niche radius* to classify all individuals into several groups by the density of some of them in the same domain with niche radius to avoid overlapping the same peak in the fitness landscape. The *dynamic niche radius* is updated each iteration step as follows:

$$m_i^{dyn} = \begin{cases} \sigma & (\text{if } m_i^{dyn} < \sigma) \\ m_i^{dyn} & (\text{otherwise}) \end{cases} \tag{15}$$

By ~~these~~ equation, the movement of solutions is given by

$$v_i^{t+1} = v_i^t + (x_i^t - x_{NR*}) * f_i \tag{16}$$

$$x_i^{t+1} = \begin{cases} x_i^t + v_i^{t+1} & (\text{if } d_{ij}^t < m_i^{dyn}) \\ x_i^t & (\text{otherwise}) \end{cases} \tag{17}$$

Algorithm 2 Dynamic Niche Radius

```
Require: Current Population x_i(i = 1, 2, ..., N) and v_i
    for i=1 to N do
2:    for j=1 to N do
        Calculate d_ij between individuals i, j
4:      if ( d_ij < σ) then
          sh(d_ij) = (1 − d_ij/σ) [eq.(10)]
6:      else
          sh(d_ij) = 0 [eq.(10)]
8:      end if
      end for
10:   m_i = Σ_{j=1}^{N} sh(d_ij) [eq.(11)]
    end for
12: for i=1 to N do
      if (m_i < σ) then
14:     m_i^dyn = σ [eq.(15)]
      else
16:     m_i^dyn = m_i [eq.(15)]
      end if
18: end for
    return Dynamic Niche Radius m_i^dyn
```

where x_{NR*} is the best solution in the domain of niche radius and d_{ij}^t indicates the euclidean distance between the nearest neighbor solutions.

In the local search phase, the new solution x_{loc} is generated around the best solution x_{NR*} in the domain of the radius as follows:

$$x_{loc} = x_{NR*} + A_i^t * rand(1, D, [-m_i, m_i]) \tag{18}$$

where A_i ~~is~~ the same values as BA. In the global search phase, the new solution x_{rnd} is generated randomly in all domains of the radius as follows:

$$x_{rnd} = x_i^t + rand(1, D, [-m_i, m_i]) \tag{19}$$

4.2 Algorithm Description

The pseudo codes of Dynamic Niche Radius and DNRBA are given in Algorithm 2 and 3. Its brief summary is described below.

- STEP1: Population initialization of bats (line 1 to 3) The population of bats $x_i(i = 1, 2, ..., N)$, the loudness A_i^0, the pulse rate r_i^0 are initialized as the initial values. The frequency f_i is initialized by Eq.(1).
- STEP2: Calculate dynamic niche radius m_i^{dyn}(line 5)
- STEP3: New solution updates (line 7) The new solutions x_i ~~is~~ calculated by Eqs. (2)(3).
- STEP4: New solution generation around the best solution x_{NR*} (line 8 to 11) A new solution x_{loc} is generated around x_* by Eq. (4) when the pulse emission rate r_i is lower than a random value.
- STEP5: Random new solution generation (line 11) A new solution x_{rnd} is generated randomly by Eq. (5).
- STEP6: Solutions update (line 15 to 16) When $rand < A_i$, the best solution is selected from x_i, x_{loc}, and x_{rnd} by Eqs.(6),(7)

Algorithm 3 Bat Algorithm with Dynamic Niche Radius (DNRBA)

Require: Objective Function $F(x)$
  Initialize Population $x_i (i = 1, 2, ..., N)$ and v_i
  Define frequency f_i at location x_i [eq.(1)] IE
  Initialize pulse rates r_i, and loudness A_i
  while (t < Max number of iterations) **do**
    Calculate Dynamic Niche Radius (Algorithm 2)
    for i=1 to N **do**
      Generate a new solution x_i and velocity v_i [eqs.(16),(17)] IE
      if ($rand > r_i$) **then**
        Generate a new solution x_{loc} around a global best solution x_i [eq.(18)] IE
      end if
      Generate a new solution x_{rnd} randomly [eq.(19)] IE
      if ($rand < A_i \& \min(F(x_i), F(x_{loc}), F(x_{rnd}) < F(x_{i*}))$ **then**
        Accept the new solution, and update pulse rate r_i & loudness A_i [eqs. (6)(7)] IE
      end if
      Evaluate all bats and select a best solution x_* in the current solutions
    end for
  end while

- STEP7: Return to STEP2

5 EXPERIMENTAL PROCEDURE

H
H

To measure the number of local optima and the convergence speed, we compare ~~with~~ the performance of all algorithms. In this section, two multimodal functions which are considered for minimization, are employed ~~in~~ [7][1].

5.1 Multimodal Test Functions

F_1: **Griewank Function**

This function is described as follows as shown in Fig. 1.

$$F_1(x) = \sum_{i=1}^{D} \frac{x_i}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \tag{20}$$

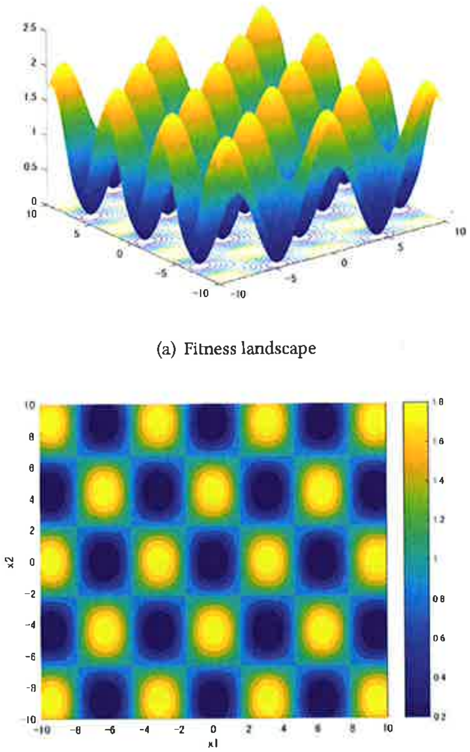
The fitness value of the global optima is $F(x_*) = 0$. This function has 17 optima (1 global optimum and 16 local optima) in the range between $x_1, x_2 \in [-10, 10]$.

F_2: **Shubert Function**

This function is described as follows as shown in Fig. 2.

$$F_2(x) = \prod_{i=1}^{D} \sum_{j=1}^{5} j \cos[(j+1)x_i + j], \tag{21}$$

where D is the number of dimension s and the fitness value of the global optima is $F(x_*) = -187.731$. This function has $D \cdot 3^D$ global optima and numerous local optima in the range of the search space ~~between~~ $x_i \in [-10, 10]^D$ with $i = 1, 2, ..., D$. H Fig. 2 shows an example of the Shubert 2D function which has 18 global optima.



(a) Fitness landscape

(b) Contour plot

Figure 1: F_1: Griewank Function

5.2 Experimental Measurement

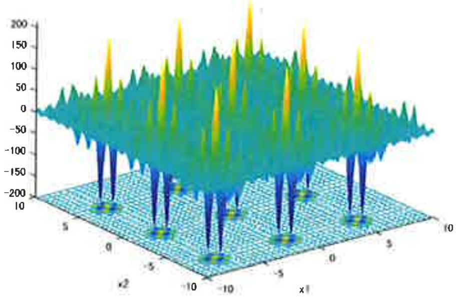
5.2.1 Peak Ratio. This experiment employs Peak Ratio(PR) [10] as the evaluation criterion in the CEC (*IEEE Congress on Evolutionary Computation*) 2013 competition [11]. The PR value measures the ratio of the found global and local optima in the total number of true peaks and it is calculated as follows: meaning of PR ?

$$PR = \frac{\sum_{run=1}^{MR} FPs}{TP * MR} \tag{22}$$

where MR indicates the maximum run, FPs indicates ~~indicates~~ the number of peaks found by the optimization algorithm. TP indicates the number of all known peaks of the function. We define that the peak is found when the Euclid ean distance between the all known peaks and the nearest solution calculated by the optimization algorithm is less than the thresholds $\varepsilon = \{1.0E-1, 1.0E-2\}$. H why 2 values?

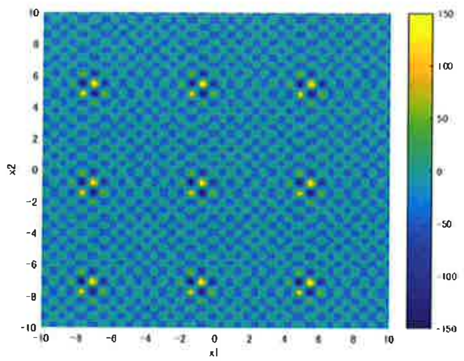
5.2.2 Peak Accuracy. To measure how ~~far~~ close solutions are ~~close~~ to the peaks (global and local optima), we employ Peak Accuracy (PA) [10] calculated as follows: defined as

$$PA = \sum_{j=1}^{TP} |F(s_j) - F(x_{NN_j})|, \tag{23}$$



(a) Fitness landscape

H that of ?



(b) Contour plot

Figure 2: F_2: Shubert Function

where s_j and x_{NN_j} denote the each known peak and the nearest neighbor solution. As the closest distance between both of them is short, the value of PA is close to 0.

5.3 Experimental Parameters

We employ the parameters as follows: frequency $f_{max} = 1$, $f_{min} = 0$, loudness $A^0 = 1$, parse rate $r^0 \in [0,\ 1]$ with $\alpha = \gamma = 0.9$. The population size $N = 100$. This experiments are ~~simulated~~ 30 runs with different random seeds and 30000 evaluations as the termination condition for each run.

H conducted with

6 RESULTS AND ANALYSIS

To test the effect of the DNRBA mechanism, this section investigates the peak ratio (PR) and the peak accuracy (PA) of each benchmark test function. Table 1 and 2 show the results that the PR and the PA values of two algorithms based on the settings of averaged over 30 individual runs at the final iteration. Fig. 3 and ?? show that the solutions relocating and exploiting at the final iteration for all functions.

?

unclear expression

6.1 Peak Ratio

The case of $\varepsilon = 1.0E-1$ from Table 1, the PR values of DNRBA were greatly higher than BA in F_1. It can be seen that DNRBA found almost all global and local optima, as shown in Fig. 4(a). Although Fig. 4(b) appears to locate solutions at all peaks, the values of DNRBA ~~was~~ less than BA in F_2, because F_2 ~~is~~ extremely sharp and complex landscape. Moreover, the exploitation of BA outperformed ~~than~~ DNRBA so that F_2 is for finding only global optima (*not searching local optima*). Table 2 shows the results of the case of $\varepsilon = 1.0E-2$. The value of DNRBA is the same as $\varepsilon = 1.0E-1$ in F_1. However, DNRBA remarkably decreased from 0.4241 to 0.0426 in F_2. It can be seen that the exploitation of DNRBA is not efficient to reach the peaks precisely.

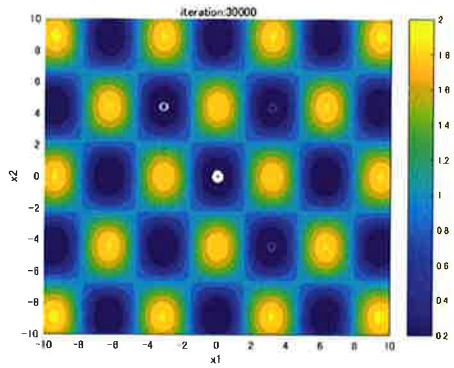
for H we has an sentence incomplete H

6.2 Peak Accuracy

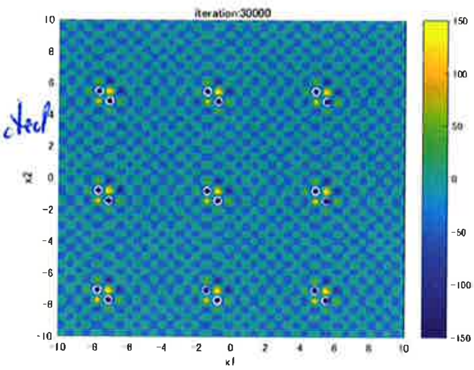
DNRBA outperformed ~~than~~ BA for F_1 from Table 1 and 2 in $\varepsilon = 1.0E-1, 1.0E-2$ to find global and local optima. In contrast, the value of BA ~~has worked well more~~ than DNRBA in F_2. BA enables to locate all solutions at all peaks from Fig. 3(b). As well as Fig. 4(b), although DNRBA appears to locate at global optima (several solutions are located at no peaks), the value of DNRBA is lower than BA.

for result was better

unclear fitness value ?



(a) F_1



(b) F_2

Figure 3: BA

5

Table 1: Peak Ratio and Peak Accuracy of BA and DNRBA (averaged over 30 runs)

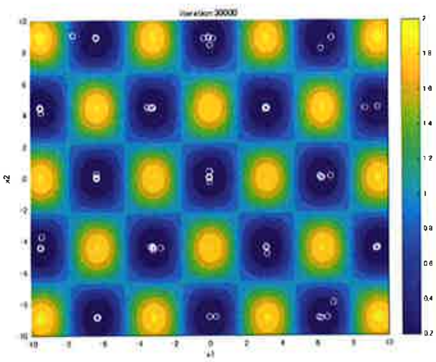
$\varepsilon = 1.0E-1$

	BA		DNRBA	
Function	PR (Mean and SD)	PA (Mean and SD)	PR (Mean and SD)	PA (Mean and SD)
F_1	0.2725 ± 0.0598	0.2416 ± 0.0149	**0.9373** ± 0.1176	**0.0094** ± 0.0155
F_2	**0.4889** ± 0.1819	**1.8160** ± 0.4990	0.4241 ± 0.388	2.4123 ± 0.6780

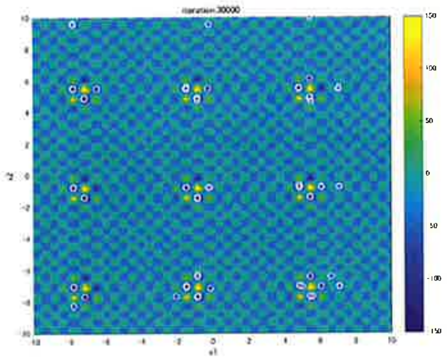
Table 2: Peak Ratio and Peak Accuracy of BA and DNRBA (averaged over 30 runs)

$\varepsilon = 1.0E-2$

	BA		DNRBA	
Function	PR (Mean and SD)	PA (Mean and SD)	PR (Mean and SD)	PA (Mean and SD)
F_1	0.2725 ± 0.0608	0.2416 ± 0.0151	**0.9373** ± 0.1176	**0.0094** ± 0.0155
F_2	**0.0556** ± 0.0619	**1.8160** ± 0.4990	0.0426 ± 0.0477	2.4123 ± 0.6780



(a) F_1



(b) F_2

Figure 4: DNRBA

7 CONCLUSION

a ed a

This paper proposes BA extending with Dynamic Niche Radius which enables to avoid overlapping the solutions into the same peak and locate multiple global optima in several ~~different~~ multimodal functions. To evaluate the performance of DNRBA, this algorithm were compared with BA. The results show that NRBA outperformed ~~to locate solutions~~ at not only global optima but also local optima in the fitness landscape, because the spatial distribution mechanism in DNRBA copes with the trade-off between the exploration and the exploitation. In contrast, BA is still better than NRBA regarding the exploitation due to converge to a single global best solution. In future work, we will compare DNRBA with current state-of-the-art algorithms and apply for dynamic optimization problems.

ing H H BA not clear enough it

REFERENCES

[1] R. Stoean et al. C. Stoean, M. Preuss. 2010. Multimodal optimization by means of a topological species conservation algorithm. *IEEE Transactions on Evolutionary Computation* 14, 6 (2010), 842–864.

[2] K.A . DeJong. 1975. An analysis of the behavior of a class of genetic adaptive system. *Ph.D. dissertation, Dept. Comput. Sci.* (1975).

[3] J. H. Holland. 1975. Adaptation in natural and artificial systems. *Ann Arbor: University of Michigan Press.* (1975).

[4] X. Li. 2005. Efficient Differential Evolution using Speciation for Multimodal Function Optimization. *In Proceedings of the Conference on genetic and evolutionary computation (GECCO'05)* (2005), 873–880.

[5] L. M. Patniak M. Srinivas. 1994. Genetic Algorithms: A Survey. *Computer* 27, 6 (1994), 17–26.

[6] B. Miller. 1996. Genetic algorithms with dynamic niche sharing for multimodal function optimization. *In: Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)* (1996).

[7] M. Molga and C. Smutnicki. Retrieved June 2013. Test functions for optimization needs (2005). (Retrieved June 2013). http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf.

[8] J. Kennedy R. Eberhart. 1995. A New Optimizer Using Particle Swarm Theory. *Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service center, Pis-cataway, NJ* 54, 1 (1995), 39–43.

[9] K. Price R. Storn. 1997. Differential Evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341–359.

[10] R. Thomsen. 2004. Multimodal Optimization using Crowding-based Differential Evolution. *In the IEEE Congress on Evolutionary Computation (CEC2004)* 2, 4 (19–23 June 2004), 1382–1389.

[11] A. Engelbrecht X. Li and M. G. Epitropakis. 2013. Benchmark Functions for CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization. *Evol. Comput.* (2013).

6