2015 IEEE Symposium Series on Computational Intelligence

Differential Evolution with Random Walk Mutation and an External Archive for Multimodal Optimization

Yu-Hui Zhang, Meng-Ting Li, Yue-Jiao Gong* and Jun Zhang
Department of Computer Science, Sun Yat-sen University, Guangzhou, China
Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education
Engineering Research Center of Supercomputing Engineering Software, Ministry of Education
*Corresponding Author: gongyuejiao@gmail.com

***Abstract*—Locating multiple optima of a problem is an important and challenging task for many real-world applications. In this paper, a random walk mutation strategy is proposed for differential evolution (DE) to handle multimodal optimization problems. The mutation strategy is able to find a balance between exploitation and exploration. First, the neighborhood and fitness information of individuals is incorporated into mutation to guide the formation of donor vectors. This facilitates the evolution of individuals toward their nearby optima. Second, the exploration ability of the mutation strategy is preserved by simulating a random walk process. Moreover, an archive technique is designed to detect converged subpopulations. The converged individuals are then reinitialized to search for other optima. This enhances the algorithm's exploration ability. Meanwhile, found optima can be maintained throughout the optimization process by using the archive technique. The random walk mutation strategy and the archive technique are integrated with DE to make a competitive multimodal algorithm. The resulting algorithm is tested on a recently proposed benchmark function set. Experimental results show that the proposed algorithm is able to provide better performance than a number of state-of-the-art multimodal algorithms.**

I. INTRODUCTION

Many optimization problems have multiple high quality solutions. These problems are commonly seen in real-world applications and are known as multimodal optimization problems. Some applications require that a problem solver finds out all the optimal solutions to a given problem [1]. In other applications, finding multiple optima of a problem is beneficial. First, when it happens that some solutions cannot be realized due to physical constraints, we can quickly switch to other high quality solutions without causing serious performance loss. Second, multiple distinct solutions may provide us some useful information about the problem being solved. Third, the probability of getting stuck in one local attraction is decreased if the goal of optimizers is to locate multiple optima.

One of the most promising approaches for solving multimodal problems is evolutionary algorithm (EA). EAs [2]-[4] are population-based stochastic optimization techniques inspired by evolutionism. EAs such as genetic algorithm (GA) [4] and differential evolution (DE) [6], are effective algorithms for solving various kinds of optimization problems. However, these algorithms are originally designed to search for a single optimal solution. Modifications must be made before they can be used to handle multimodal problems. EAs have inherent advantages over traditional optimization approaches owing to their population-based search strategy. To tackle multimodal problems, techniques known as "niching" are proposed in the literature. Famous niching techniques include crowding [7], clearing [8], restricted tournament selection [9], fitness sharing [10], and speciation [11]. They are integrated into classical EAs to solve multimodal problems. Generally, they modify the behavior of EAs such that multiple optima can be detected and maintained simultaneously.

Niching techniques can be roughly divided into two groups, i.e., generic niching techniques and specialized niching techniques. Generic niching techniques generally modify the selection mechanism of an EA so as to maintain population diversity and avoid all individuals converge to a single optimum. They can be integrated with different kinds of EAs to make a class of multimodal algorithms. Some early niching techniques (crowding, speciation, fitness sharing) belong to this category. In comparison, a specialized niching technique is tailored for a specific kind of EA. Specialized niching techniques change the way in which offspring population is generated. By using these techniques, subpopulations are formed automatically within basins of attractions. Each subpopulation will finally converge to one possible optimum. Some recently proposed multimodal algorithms ([12]-[16]) belong to this category. It is noteworthy that the two groups of niching techniques can be used together to make more powerful multimodal algorithms. For example, Qu *et al* [17] proposed a neighborhood mutation strategy and integrate it with crowding, speciation, and fitness sharing. Experimental results reported in [17] show that the integrated algorithms are very competitive.

One key challenge in designing niching techniques is to find a balance between local solution exploitation and global exploration. DE is efficient in performing global search [6]. However, when dealing with multimodal functions, it is important to enhance the algorithm's exploitation ability while at the same time preserving its exploration ability. Note that a niching DE generally divides its population into multiple subpopulations, it would make the algorithm more effective if sufficient efforts are made to exploit the inner regions bounded by the individuals of each subpopulation. Motivated by the findings, we propose a random walk mutation based DE (RW-DE) to handle multimodal functions. The strategy simulates the

This work was supported in part by the National High-Technology Research and Development Program (863 Program) of China under Grant 2013AA01A212, in part by the National Science Fund for Distinguished Young Scholars under Grant 61125205, and in part by the National Natural Science Foundation of China under Grant 6120002 and Grant 61502542.

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.260

random walk process to choose the parameter vectors (x_{r1}, x_{r2}, and x_{r3}) that used to generate donor vectors. The random walk mutation is tailored in a way that encourages each individual to evolve toward its nearby optimum. Like the neighborhood mutation strategy [17], the proposed strategy is cooperated with the crowding technique to increase the algorithm's ability of maintaining multiple optima.

To accelerate the convergence speeds of subpopulations, a local parameter adaptation method is employed. Noticing that different subpopulations may have different convergence speeds, it is a waste of computational resource if the converged subpopulations are kept taking fitness evaluations away. Therefore, an archive technique is designed to detect and store converged subpopulations. With the external archive, the proposed algorithm can not only maintain found optima until the end of the search, but also enhance the algorithm's exploration ability. Experiments have been conducted to investigate the effect of the random walk mutation strategy and the archive technique. In addition, the overall performance of the proposed algorithm on a population benchmark function set. The experimental results demonstrate the effectiveness of the proposed algorithm.

The rest of this paper is organized as follows. Section II reviews the basic DE and its variants for multimodal optimization. Section III gives a detailed description of the proposed algorithm. Experimental setup and numerical results are presented in Section IV. Finally, concluding remarks are given in Section V.

II. Background

A. *Differential Evolution*

Differential evolution (DE) [6] is a simple yet powerful optimization technique proposed by Storn and Price. It is efficient in solving various global optimization problems [18]. Like other EAs, DE is a population-based stochastic search technique. It maintains a population of *Np* individuals (candidate solutions) during the search process. Unlike other EAs, DE mutates individuals by using the differences between randomly sampled pairs of individuals from the population. DE has four main steps, i.e., initialization, mutation, recombination, and selection. The last three steps are repeated until the termination criterion is satisfied.

DE starts by scattering *Np* individuals $\{x_i \mid i=1,2,\ldots,Np\}$ over the search space. At each iteration, for each target vector x_i, we generate a donor vector v_i using the following formula:

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

where $r1$, $r2$, and $r3$ are mutually exclusive integers chosen from $\{1, 2, \ldots, Np\}\backslash i$. F is a scale factor. Then, we generate a trial vector by combining the components of target vector x_i and donor vector v_i:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } randj \le Cr \text{ or } j = k \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{2}$$

where *Cr* is the crossover rate that decides whether a component should come from x_i or v_i. The selection process is based on a comparison between the trial vector u_i and the target vector x_i. If the fitness value of u_i is better than that of x_i, then u_i will take the place of x_i in the next iteration.

DE is a promising candidate for multimodal optimization. Some generic niching techniques have been incorporated into DE to tackle multimodal problems. In addition, there are niching techniques tailored for DE.

B. *Generic Niching Techniques and Their Integration with DE*

1) Crowding: Crowding method, proposed by De Jong in 1975 [19], is one of the earliest niching techniques. The idea is that similar individuals in the population must compete with one other for limited resources. The similarity is generally measured by the Euclidean distance between individuals. An offspring is compared with *CF* randomly sampled individuals in the population, where *CF* is an integer parameter called crowding factor. The nearest rival will be replaced if the offspring has better fitness. This approach is advantageous to the diversity maintenance. However, replacement errors occur when we do the sampling. To reduce the replacement error, a deterministic crowding method is later proposed by Mahfoud [20]. It eliminates the parameter *CF*. An offspring is compared with the nearest individuals in the population. In a later study, Thomsen [7] proposed to extend DE with the crowding mechanism (CDE) to tackle multimodal problems.

2) Speciation: The idea of speciation [11] is to divide the population into a number of species according to the distance information. Each species is formed around a dominated individual called species seed. A species seed is first identified. Then, individuals whose distance to the species seed is less than a threshold value r_s are assigned to the species. The threshold value r_s (called niche radius) controls the range of every species. The evolutionary operators of EAs (e.g. crossover and mutation) are performed within each species. In a later study, Li [21] proposed a speciation-based DE (SDE) applying the above speciation concept.

*3) Fitness sharing:*The fitness sharing method is an effective niching technique proposed by Holland [22]. It is later extended by Goldberg and Richardson [10]. The idea is that an individual should share information with other individuals that are close to it. Specifically, the shared fitness of the *i*th individual is computed as follows:

$$shfit(i) = \frac{fit(i)}{\sum_{j=1}^{Np} sh(d_{ij})} \tag{3}$$

where the sharing function is defined as:

$$sh(d_{ij}) = \begin{cases} 1 - (d_{ij} / \sigma_{\text{share}})^{\alpha}, & \text{if } d_{ij} < \sigma_{\text{share}} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where d_{ij} is the distance between individuals x_i and x_j. σ_{share} is a threshold value called sharing radius. α is a constant called sharing level. In a later study, Thomsen [7] integrated the fitness sharing concept with DE and proposed a sharing DE (ShDE).

C. Specialized Niching Techniques Tailored for DE

Most of the specialized niching techniques concentrate on distributing offspring over a number of niches. This mechanism can help exploit the region of attraction that has been detected by the parental population. Some of the niching techniques are described in this subsection. Epitropakis *et al.* [15] proposed two mutation strategies to handle multimodal functions. For target vector x_i, instead of choosing the base vector randomly, the individual closest to x_i is selected. The strategies are called DE/nrand/1 and DE/nrand/2 respectively. They can be formulated as follows:

$$v_i = x_{NNi} + F \cdot (x_{r1} - x_{r2}) \tag{5}$$

$$v_i = x_{NNi} + F \cdot (x_{r1} - x_{r2}) + F \cdot (x_{r3} - x_{r4}) \tag{6}$$

where x_{NNi} is the closest individual to x_i. The donor vector generated by the formula is likely to be in the vicinity of x_i. Hence, the strategies are beneficial to the preservation of population distribution. In the follow-up work, Epitropakis *et al.* [16] combine the first mutation strategy (DE/nrand/1) with an adaptive parameter control method and a dynamic archive technique to make a more competitive multimodal algorithm. The resulting algorithm, called dADE/nrand/1, turns out to be very effective.

To improve the niching performance of existing DE-based multimodal algorithms, Qu *et al.* [17] proposed a neighborhood mutation strategy. In the strategy, the generation of a donor vector is limited to the neighborhood of its corresponding target vector. More specifically, for target vector x_i, the parameter vectors (x_{r1}, x_{r2}, and x_{r3}) are randomly chosen from the *m* nearest neighbors of x_i, instead of from the entire population. The parameter *m* is used to control the neighborhood size. It is suggested to set between 1/20 and 1/5 of the population size *Np*. The mutation strategy is then integrated with three existing algorithms, i.e., CDE, SDE, and ShDE. Experimental results reported in [17] show that the neighborhood mutation based algorithms (NCDE, NSDE, and NShDE) significantly outperform the original algorithms.

From the above descriptions, it can be observed that the way of choosing parameter vectors (x_{r1}, x_{r2}, and x_{r3}) plays a very important part on DE's niching performance. This paper is devoted to finding a better way of choosing parameter vectors that will make DE more effective in multimodal optimization. To this end, we propose a random walk mutation strategy. Different from aforementioned methods, the mutation strategy simulates a random process while choosing the parameter vectors. It not only takes into the localities of individuals, but also their fitness.

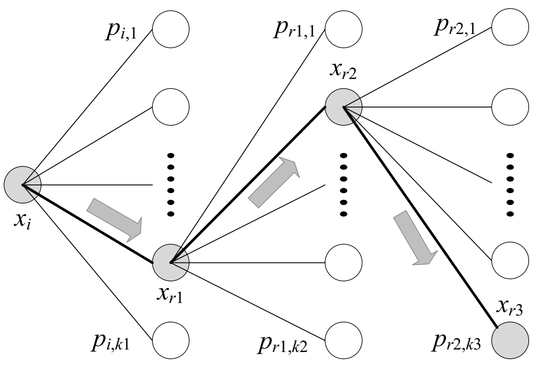


Fig. 1. Illustration of the random walk mutation strategy.

III. The Proposed Algorithm

A. Random Walk Mutation

In the standard DE, several parameter vectors are randomly picked from the entire population to generate a donor vector v_i. If the parameter vectors are far from one another, the length of the difference vector will be very large. This prevents premature local convergence and enhances DE's global search ability. However, this strategy is not efficient when dealing with multimodal functions, since multiple optima need to be located simultaneously. To facilitate multiple convergence, a random walk mutation strategy is proposed in this paper. We regard the process of choosing parameter vectors as a random walk. Here, a random walk refers to a path that consists of a series of successive random steps. The term random walk was introduced by Karl Person [23] in 1905 and it serves as a fundamental model for many stochastic activities in different fields (e.g., ecology, economics, computer science, and physics).

The random walk mutation strategy is illustrated in Fig. 1. Starting from the target vector x_i, we take random steps successively to choose parameter vectors. More specifically, the vector x_{r1} is chosen from x_i's *k*1 nearest neighbors, where *k*1 is a random integer between 5%*Np* and 25%*Np*. Each neighbor $n_{i,j}$ ($j \in [1, k1]$) of x_i is assigned a selection probability, which is calculated as:

$$P_{i,j} = \frac{nfit(n_{i,j})}{\sum_{m=1}^{k1} nfit(n_{i,m})} \tag{7}$$

where $nfit(n_{i,j})$ is the normalized fitness of the *i*th neighbor. It is computed as:

$$nfit(n_{i,j}) = \frac{fit(n_{i,j}) - worstfit}{bestfit - worstfit} \tag{8}$$

In (8), *bestfit* and *worstfit* are the best and worst fitness values in the current population respectively. x_{r1} is chosen using a roulette wheel based on the probabilities. By analogy, x_{r2} is chosen from x_{r1}'s *k*2 nearest neighbors and x_{r3} is chosen from x_{r2}'s *k*3 nearest neighbors. *k*2 and *k*3 are two random integers within [5%*Np*, 25%*Np*].

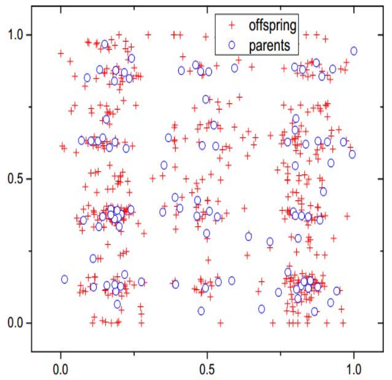


Fig. 2. Offspring distribution using the random walk mutation.

The mutation strategy makes use of both fitness and neighborhood information to select the parameter vectors. On one hand, the donor vector is very likely to be in the vicinity of its target vector. The offspring (trial vector) will have more chances to approach the optimum close to its parent. This facilitates the exploitation of candidate regions located by the parental population. On the other hand, the donor vector can also 'walk' very far from the target vector. The offspring will have some chances to explore "new continents". This preserves DE's exploration ability and it would make the algorithm more effective in complex high dimensional multimodal functions. Hence, the strategy is able to find a balance between the algorithm's exploitation and exploration abilities. The effect of the mutation strategy is demonstrated in Fig. 2, where blue circles represent the parental population. The parental population consists of 100 individuals. It can be seen that clustering tendency has emerged in the parental population. 500 trial vectors (denoted by red crosses) are generated by using the random walk mutation strategy. From the figure, it can be seen that the distribution of the parental population is well preserved by the offspring. Most of the offspring tend to exploit the promising regions located by their parents. Meanwhile, efforts have also been paid to explore some unknown areas.

B. *Local Parameter Adaption*

The population is divided into multiple subpopulations so as to locate multiple optima. The suitable scale factors *F* for the subpopulations may vary. Therefore, a local adaptive parameter tuning method is developed to find the suitable scale factors and to increase the convergence speed of subpopulations. The method is a modification of the adaptation scheme proposed in [24]. It is specially designed for the parameter tuning of multimodal algorithms. Instead of using a location parameter of the Cauchy distribution for the whole population, each individual maintains its own μ_{Fi}. The scale factor for target vector x_i is generated using the following formula:

$$Fi = \mathrm{Cauchy}(\mu_{Fi}, 0.1) \tag{9}$$

The parameter μ_{Fi} is updated at the end of each generation as:

$$u_{Fi} = 0.9 \cdot u_{Fi} + 0.1 \cdot \mathrm{mean}_L(S_{Fi}) \tag{10}$$

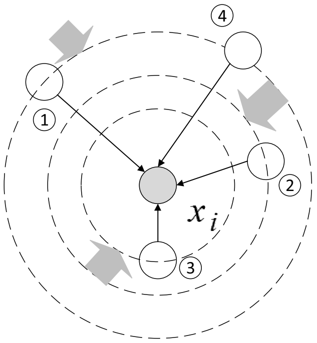


Fig. 3. Detection of a stagnant individual.

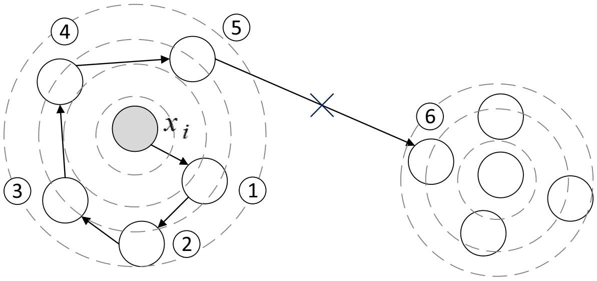


Fig. 4. Detection of a converged subpopulation.

where $\mathrm{mean}_L(S_{Fi})$ is the Lehmer mean of the elements in set S_{Fi}. S_{Fi} contains successful *F*s that belong to the 5%*Np* nearest neighbors of x_i. Here, *Fi* is called a successful scale factor if the trial vector generated by *Fi* has successfully replaced a target vector.

C. *Archive Technique*

Similarly, the convergence speeds of the subpopulations may vary. It would degrade the performance of multimodal algorithms if the subpopulation that have already converged are taking up computational resources (fitness evaluations). To make good use of limited fitness evaluations, an archive technique is developed: The technique is used to detect converged subpopulation. Once a converged subpopulation is recognized, it is stored in an external archive and then reinitialized to search for other optima. This mechanism not only avoid the waste of fitness evaluations, but also enhance the algorithm's exploration ability. Besides, the utilization of the external archive ensures that optima found in previous generations can be maintained throughout the optimization process.

The process of detecting a converged subpopulation is described as follows. For each individual x_i in the population, we maintain a stagnation counter and a radius *R* that represents the range of x_i's territory. The stagnation counter (i.e., the number of comparisons conducted since x_i's last improvement) is initialized to 0 and the radius is set to infinite. After a trial vector u_j has been generated, it is compared with the nearest individual in the population. Suppose that x_i is the individual nearest to u_j. If u_i is better than x_i, then u_j takes the place of x_i and its stagnation counter and radius are set to 0 and infinite

Algorithm 1 RW-DEAR-LP

```
$G$=0;
Generate an initial population $P_0=\{x_1, x_2, \ldots, x_{Np}\}$ by randomly sampling from the search space;
Evaluate the fitness values of the individuals;
$FEs=NP$;
$AR=\emptyset$;
while $FEs<MaxFEs$ do
  for $x_i$ in $P_G$ do
    Use the random walk mutation strategy, with the control parameter generated using (9), to generate a trial vectors $u_i$;
    Evaluate the fitness values of the trial vectors $u_{i,}$;
    $FEs=FEs+1$;
    Compare $u_i$ with the nearest individual $x_j$ in the $P_G$;
    if $u_i$ is better than $x_j$ then
      Replace $x_j$ with $u_i$;
    else if dis($u_i$, $x_j$)<$x_j.R$ then
      $x_j.stag = x_j.stag+1$; //$x_j.stag$: stagnation counter
      $x_j.R$ = dis($u_i$, $x_j$); // $x_j.R$: radius of $x_j$'s territory
      if $x_j.stag>\theta$ then
        Attach the label 'converged' to the individuals that belong to the same subpopulation as $x_j$;
      end if
    end if
  end for
  for $x_i$ in $P_G$ do
    if $x_i$ is labeled 'converged' then
      Add $x_i$ to the archive $AR$;
      Reinitialize $x_i$;
    else
      Update $\mu_{Fi}$ using (10);
    end if
  end for
  $G=G+1$;
end while
```

respectively. Otherwise, x_i prevails. In this case, we calculate the distance d_{ij} between x_i and u_j. If d_{ij} is less than the radius R (i.e., u_j intrudes into x_i's territory), then we narrow the territory of x_i down to a circle (or a hyper-sphere) of radius $R=d_{ij}$ and increase x_i's stagnation counter by one. If d_{ij} is larger than R, u_j is out of x_i's territory and thus nothing happens to the stagnation counter. Once the stagnation counter exceeds a certain threshold θ, x_i is judged to be stagnant. The territory of x_i will become smaller and smaller until x_i is replaced by a trial vector or x_i is stagnant. The mechanism of the shrinking territory makes the detection process very stable. Fig. 3 records the changes of the stagnation counter and the territory of a target vector x_i. The dash circles represent the territory of x_i. Four trial vectors are compared with x_i in serial order. After the comparisons, the stagnation counter becomes three and x_i's territory becomes the smallest dash circle. The 4th trial vector does not affect x_i's stagnation counter since it is out of x_i's territory at the time the comparison is conducted.

The detection of a converged subpopulation takes a number of steps after the finding of a stagnant individual. Fig. 4 illustrates the procedures. The grey dash lines are used to denote contours of the problem landscape. For a stagnant individual x_i (represented by the grey circle), we first find out its nearest neighbor (the No. 1 individual) in the population. Then, a hill-valley method [25] is used to test whether there is a valley between x_i and its nearest neighbor. If the answer is no, then x_i and its neighbor belong to a same peak. The No. 1 individual is labelled "converged". Next, we proceed to find the nearest unlabeled neighbor of the No. 1 individual and repeat the above steps. The process ends when we reach an individual (the No. 6 individual in the example) that belongs to a different niche. This way, a converged subpopulation is recognized. At the end of each iteration, the converged subpopulations are stored in an archive and reinitialized to search for other optima.

D. Random Walk Mutation-based DE with an External Archive and Local Parameter Adaptation (RW-DEAR-LP)

The pseudo of the algorithm that incorporates the above components is presented in **Algorithm 1**. At each iteration, the selection of the parameter vectors takes $O(Np^2)$ time. The detection of converged subpopulation takes no more than $O(Np)$ time. Therefore, the total time complexity of the algorithm is $\text{O}(Np^2)$. It is important to note that distance information is necessary for inducing stable niching behavior. Due to the calculation of distances, the time complexities of most of the multimodal algorithms described in Section II are $O(Np^2)$. From this viewpoint, the random walk mutation does not impose serious burden on the algorithm's complexity.

IV. Experiments

In this section, we carry out experiments to investigate the effect of the random walk mutation strategy and the archive technique. Further, the integrated algorithm (RW-DEAR-LP) is compared with a number of state-of-the-art multimodal algorithms to demonstrate its performance.

A. Experimental Setup

1) Test Functions: The CEC2013 benchmark function set [26] is adopted in this paper to study the performance of multimodal algorithms. The function set contains 12 multimodal functions. F_1-F_5 are simple low-dimensional multimodal functions with a small number of optima. F_6-F_8 are scalable multimodal functions with a large number of optima. In particular, the number of optima for F_6 and F_7 is determined by the dimensionality. F_9-F_{12} are scalable composite multimodal functions. They are much more complex than those basic functions. Detailed descriptions of the function set can be found in [26].

2) Parameter Settings: An algorithm terminates when the given number of fitness evaluations (FEs) is exhausted. The maximum numbers of FEs for the test functions are provided

Table I
MaxFEs Used For 3 Ranges of Test Functions

Range of functions	MaxFEs
F_1 to F_5 (1D or 2D)	5.00E+04
F_6 to F_{11} (2D)	2.00E+05
F_6 to F_{12} (3D or higher)	4.00E+05

TABLE II

EXPERIMENTAL RESULTS OF RW-DE AND DE/NARN/1

Level of	1.00E-01				1.00E-02				1.00E-03				1.00E-04				1.00E-05			
accuracy	DE/nrand/1		RW-DE		DE/nrand/1		RW-DE		DE/nrand/1		RW-DE		DE/nrand/1		RW-DE		DE/nrand/1		RW-DE	
Function	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_2(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_3(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_4(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_5(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_6(2D)	0.611	0.000	**1.000**	1.000	0.611	0.000	**1.000**	1.000	0.611	0.000	**0.952**	0.780	**0.611**	0.000	0.333	0.080	0.000	0.000	0.000	0.000
F_7(2D)	0.387	0.000	**0.698**	0.000	0.387	0.000	**0.674**	0.000	0.386	0.000	**0.672**	0.000	0.383	0.000	**0.669**	0.000	0.376	0.000	**0.666**	0.000
F_6(3D)	0.211	0.000	**0.555**	0.000	0.211	0.000	**0.535**	0.000	0.211	0.000	**0.492**	0.000	0.211	0.000	**0.380**	0.000	0.211	0.000	**0.252**	0.000
F_7(3D)	0.104	0.000	**0.277**	0.000	0.104	0.000	**0.268**	0.000	0.104	0.000	**0.268**	0.000	0.102	0.000	**0.268**	0.000	0.099	0.000	**0.267**	0.000
F_8(2D)	0.998	0.980	**1.000**	1.000	0.998	0.980	**1.000**	1.000	0.998	0.980	**1.000**	1.000	0.998	0.980	**1.000**	1.000	0.998	0.980	**1.000**	1.000
F_9(2D)	0.667	0.000	**0.967**	0.800	0.667	0.000	**0.910**	0.560	0.667	0.000	**0.880**	0.460	0.667	0.000	**0.867**	0.420	0.667	0.000	**0.853**	0.380
F_{10}(2D)	0.845	0.180	**0.918**	0.420	**0.833**	0.160	0.783	0.080	**0.805**	0.100	0.660	0.000	**0.805**	0.100	0.603	0.000	**0.793**	0.080	0.545	0.000
F_{11}(2D)	0.667	0.000	**0.687**	0.000	0.667	0.000	**0.680**	0.000	0.667	0.000	0.667	0.000	0.663	0.000	0.663	0.000	**0.663**	0.000	0.657	0.000
F_{11}(3D)	0.667	0.000	0.667	0.000	0.667	0.000	0.667	0.000	0.663	0.000	**0.667**	0.000	0.663	0.000	**0.667**	0.000	0.663	0.000	**0.667**	0.000
F_{12}(3D)	**0.505**	0.000	0.470	0.000	**0.498**	0.000	0.425	0.000	**0.480**	0.000	0.340	0.000	**0.460**	0.000	0.320	0.000	**0.438**	0.000	0.320	0.000
F_{11}(5D)	0.593	0.000	**0.660**	0.000	0.587	0.000	**0.657**	0.000	0.573	0.000	**0.653**	0.000	0.553	0.000	**0.653**	0.000	0.540	0.000	**0.653**	0.000
F_{12}(5D)	**0.360**	0.000	0.275	0.000	**0.325**	0.000	0.263	0.000	**0.305**	0.000	0.260	0.000	**0.285**	0.000	0.258	0.000	**0.260**	0.000	0.255	0.000
F_{11}(10D)	0.380	0.000	**0.497**	0.000	0.323	0.000	**0.497**	0.000	0.287	0.000	**0.493**	0.000	0.247	0.000	**0.493**	0.000	0.217	0.000	**0.493**	0.000
F_{12}(10D)	0.220	0.000	**0.335**	0.000	0.180	0.000	**0.298**	0.000	0.148	0.000	**0.275**	0.000	0.140	0.000	**0.268**	0.000	0.130	0.000	**0.253**	0.000
F_{12}(20D)	0.200	0.000	**0.285**	0.000	0.180	0.000	**0.268**	0.000	0.155	0.000	**0.263**	0.000	0.138	0.000	**0.263**	0.000	0.133	0.000	**0.263**	0.000

TABLE III

EXPERIMENTAL RESULTS OF RW-DE AND RW-DEAR

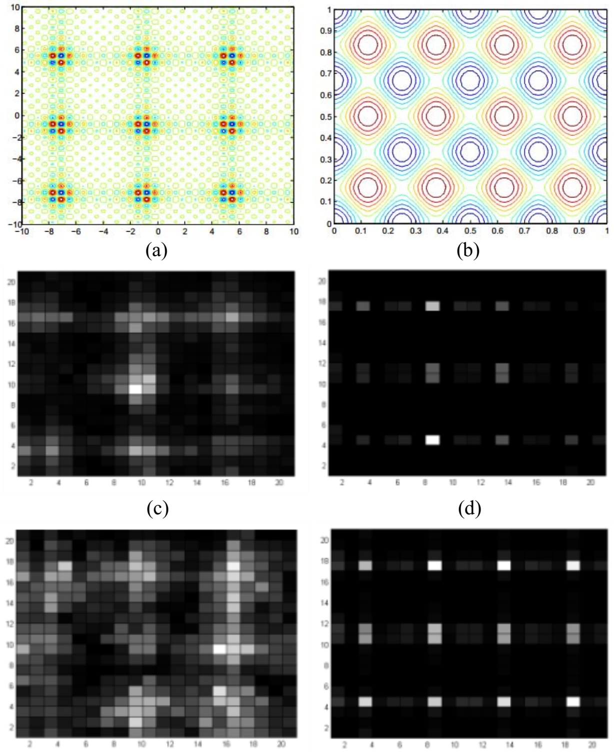
Level of	1.00E-01				1.00E-02				1.00E-03				1.00E-04				1.00E-05			
accuracy	RW-DE		RW-DEAR		RW-DE		RW-DEAR		RW-DE		RW-DEAR		RW-DE		RW-DEAR		RW-DE		RW-DEAR	
Function	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_2(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_3(1D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_4(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_5(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_6(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.952	0.780	**0.993**	0.940	0.333	0.080	**0.804**	0.540	0.000	0.000	0.000	0.000
F_7(2D)	0.698	0.000	**1.000**	1.000	0.674	0.000	**0.957**	0.200	0.672	0.000	**0.954**	0.180	0.669	0.000	**0.948**	0.120	0.666	0.000	**0.928**	0.060
F_6(3D)	**0.555**	0.000	0.534	0.000	**0.535**	0.000	0.506	0.000	**0.492**	0.000	0.473	0.000	0.380	0.000	**0.389**	0.000	**0.252**	0.000	0.238	0.000
F_7(3D)	0.277	0.000	**0.988**	0.620	0.268	0.000	**0.480**	0.000	0.268	0.000	**0.474**	0.000	0.268	0.000	**0.466**	0.000	0.267	0.000	**0.443**	0.000
F_8(2D)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
F_9(2D)	0.967	0.800	**1.000**	1.000	0.910	0.560	**0.993**	0.960	0.880	0.460	**0.930**	0.620	0.867	0.420	0.867	0.360	**0.853**	0.380	0.800	0.160
F_{10}(2D)	**0.918**	0.420	0.910	0.400	**0.783**	0.080	0.738	0.040	**0.660**	0.000	0.545	0.000	**0.603**	0.000	0.393	0.000	**0.545**	0.000	0.248	0.000
F_{11}(2D)	0.687	0.000	**0.960**	0.760	0.680	0.000	**0.783**	0.020	0.667	0.000	**0.713**	0.000	0.663	0.000	**0.690**	0.000	0.657	0.000	**0.680**	0.000
F_{11}(3D)	0.667	0.000	**0.993**	0.960	0.667	0.000	0.667	0.000	0.667	0.000	0.667	0.000	0.667	0.000	0.667	0.000	**0.667**	0.000	0.657	0.000
F_{12}(3D)	0.470	0.000	**0.983**	0.900	0.425	0.000	**0.495**	0.000	0.340	0.000	**0.390**	0.000	0.320	0.000	**0.375**	0.000	0.320	0.000	**0.368**	0.000
F_{11}(5D)	0.660	0.000	**1.000**	1.000	0.657	0.000	**0.667**	0.000	0.653	0.000	**0.667**	0.000	0.653	0.000	**0.667**	0.000	0.653	0.000	**0.663**	0.000
F_{12}(5D)	0.275	0.000	**0.688**	0.100	0.263	0.000	**0.323**	0.000	0.260	0.000	**0.300**	0.000	0.258	0.000	**0.278**	0.000	0.255	0.000	**0.268**	0.000
F_{11}(10D)	0.497	0.000	**1.000**	1.000	0.497	0.000	**0.667**	0.000	0.493	0.000	**0.667**	0.000	0.493	0.000	**0.667**	0.000	0.493	0.000	**0.657**	0.000
F_{12}(10D)	0.335	0.000	**0.435**	0.020	0.298	0.000	**0.300**	0.000	0.275	0.000	**0.278**	0.000	**0.268**	0.000	0.265	0.000	**0.253**	0.000	0.193	0.000
F_{12}(20D)	0.285	0.000	**0.535**	0.020	0.268	0.000	**0.413**	0.000	0.263	0.000	**0.413**	0.000	0.263	0.000	**0.398**	0.000	**0.263**	0.000	0.200	0.000

in Table I [26]. In addition to the multimodal algorithms described in Section II, several PSO-based multimodal algorithms (r2pso, r3pso [12], FERPSO [13], LIPS [14]) are also compared in this paper. The parameters of the compared algorithms are set according to the literature. The population size *Np* is fixed at 100 for all the test algorithms. For DE-based multimodal algorithms, *Cr* and *F* are set to 0.9 and 0.5 respectively. For RW-DEAR-LP, the threshold value θ of the archive technique is empirically set to 10.

3) Performance Measure: Two popular performance measures for multimodal optimization, i.e., peak ratio (PR) and success rate (SR), are adopted. PR is the percentage of successful located peaks and SR is the percentage of successful runs [26]. To determine whether a peak is located, we need to specify an accuracy level ε. Five levels of accuracy {1E-01, 1E-02, 1E-03, 1E-04, 1E-05} are used in the following experiment. Each algorithm runs 50 times for each multimodal function.

B. Comparison with DE/nrand/1

We first compare the random walk mutation-based DE (RW-DE) with DE/nrand/1. The PRs and SRs of the algorithms under five accuracy levels are presented in Table II. The better results are marked in bold. RW-DE outperforms DE/nrand/1 in most of the test functions in terms of PR regardless of the setting of ε. The random walk mutation strategy offers many advantages to the search of multimodal landscapes. It can exploit multiple regions effectively since both neighborhood and fitness information is incorporated into the selection of parameter vectors. The performance of RW-DE on high-



(a) (b)

(c) (d)

(e) (f)

Fig. 5. Effect of the external archive. (a) Contour plot of F_6(2D) (b) Contour plot of F_8(2D). (c)-(f): percentage of FEs spent on each grid. (c) RW-DE F_6(2D). (d) RW-DE F_8(2D). (e) RW-DEAR F_6(2D). (f) RW-DEAR F_8(2D).

dimensional multimodal functions (F_{11}-F_{12}) is also quite well, owing to the exploration ability provided by the random walk.

C. *Effect of the archive*

In this part, we study the effect of the proposed archive technique. The archive integrated version of the proposed algorithm (RW-DEAR) is compared with the one without (RW-DE). The results are shown in Table III, with better results marked in bold. It can be seen that the archive technique can improve the algorithm's performance in most of the test functions, especially when the accuracy level is set to 1E-01 and 1E-02. To demonstrate the effect of the archive, we divided the search space into 20×20 grids and record the number of FEs spent on each grid. The experiment is conducted on F_6(2D) and F_8(2D). The contour plots of the function landscapes are given in Fig. 5 (a)-(b) while the results of RW-DE and RW-DEAR are shown in Fig. 5 (c)-(d) and (e)-(f) respectively. The brightness of a grid represents the percentage of FEs spent on it. It can be seen that RW-DEAR can search the problem spaces more evenly than RW-DE. This is because by using the archive technique, converged subpopulations are detected and reinitialized to search for other optima. Hence, RW-DEAR can avoid the waste of FEs on those fully exploited regions and explore the search space more thoroughly.

D. *Overall Performance*

We continue to study the performance of RW-DEAR-LP. The algorithm is compared with a number of state-of-the-art multimodal algorithms. The accuracy level ε is fixed at 1E-04. The peak ratios obtained by the algorithms are listed in Table IV. For each test function, the ranks of the multimodal algorithms are provided in parentheses. The total ranks (summation of the individual ranks) are given in the last row of the table. From Table IV, it can be seen that the results obtained by RW-DEAR-LP are very competitive. RW-DEAR-LP ranks first in most of the test functions. Its superiority is evident when dealing with high-dimensional composite multimodal functions (F_9-F_{12}). Fig. 6 gives the convergence speeds (number of optima found versus FEs) of the multimodal algorithms on F_{11}-F_{12}. RW-DEAR-LP can locate more optima using less FEs. In addition, the overall performance (total rank) of RW-DEAR-LP turns out to be the best. The performance of RW-DEAR-LP owes to the fine search ability provided by the random walk mutation strategy. Meanwhile, the preservation of the exploration ability and the use of an external archive guarantee the algorithm's high performance in complex multimodal functions.

V. Conclusion

A random walk mutation strategy was proposed in this paper for DE to handle multimodal problems. The mutation strategy is distinguished by the way in which parameter vectors are chosen and is able to find a balance between the algorithm's exploitation and exploration abilities. Both neighborhood and fitness information is incorporated in the strategy to enhance the algorithm's fine search ability. Meanwhile, the algorithm's exploration ability is preserved by performing a random walk process. To accelerate the convergence speed of subpopulations, a local parameter adaptation method is employed. Further, an archive technique was developed to detect converged subpopulations. To avoid wasting limited fitness evaluations, the converged subpopulations are then reinitialized to search the problem space thoroughly. Experiments conducted in the paper show that the proposed algorithm, RW-DEAR-LP, is able to exhibit high performance in many multimodal problems.

References

[1] E. Cuevas, M. González, D. Zaldívar, and M. Pérez-Cisneros, "Multi-ellipses detection on images inspired by collective animal behavior," *Neural Computing and Applications*, vol. 24, no. 5, pp. 1019-1033, 2014

[2] T. Back, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. Oxford, U.K.: Oxford Univ. Press, 1997.

[3] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, and J.-J. Li, "Distributed Evolutionary Algorithms and Their Models: A Survey of the State-of-the-Art," *Applied Soft Computing*, vol. 34, pp. 286-300, 2015.

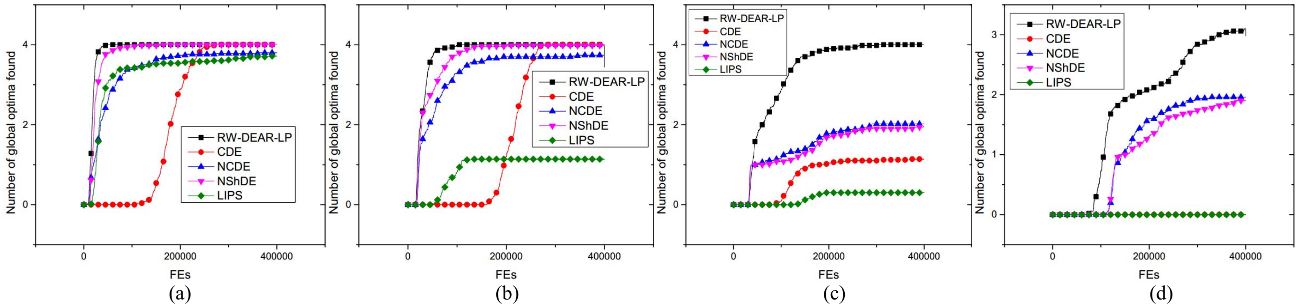
[4] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H.S.-H. Chung, Y.-H. Shi, and J. Zhang, "Genetic Learning Particle Swarm Optimization," *IEEE Trans. on Cybern.*, In Press.

[5] D. Whitley,"A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, pp. 65–85, 1994.

[6] R. Storn and K. V. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimization*, vol. 11, no. 4, pp. 341–359, 1995.

TABLE IV
COMPARISON WITH STATE-OF-THE-ART MULTIMODAL ALGORITHMS

Algorithm	r2pso	r3pso	LIPS	FERPSO	CDE	SDE	ShDE	NCDE	NSDE	NShDE	dADE/nrand/1	RW-DEAR-LP
F_1(1D)	**1(1)**	**1(1)**	**1(1)**	0.5(11)	**1(1)**	0.96(10)	0.5(11)	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**
F_2(1D)	**1(1)**	**1(1)**	**1(1)**	0.992(10)	**1(1)**	**1(1)**	0.636(12)	**1(1)**	0.984(11)	**1(1)**	**1(1)**	**1(1)**
F_3(1D)	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**
F_4(2D)	**1(1)**	0.995(7)	**1(1)**	0.76(10)	0.98(9)	0.6(11)	0.025(12)	**1(1)**	**1(1)**	0.985(8)	**1(1)**	**1(1)**
F_5(2D)	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**	0.51(12)	0.82(11)	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**
F_6(2D)	0.563(5)	0.708(4)	0.722(3)	0.331(7)	0.139(9)	0.502(6)	0(12)	0.028(10)	0.27(8)	0.009(11)	0.984(2)	**1(1)**
F_7(2D)	0.416(7)	0.375(8)	0.479(6)	0.178(10)	0.726(4)	0.202(9)	0.03(12)	0.688(5)	0.13(11)	0.826(2)	0.823(3)	**0.951(1)**
F_6(3D)	0.010(11)	0.109(6)	0.199(5)	0.055(9)	0.549(3)	0.074(7)	0(12)	0.503(4)	0.057(8)	0.044(10)	0.967(2)	**0.981(1)**
F_7(3D)	0.067(8)	0.097(7)	0.125(6)	0.029(9)	0.273(4)	0.027(10)	0.003(12)	0.262(5)	0.022(11)	0.374(3)	0.432(2)	**0.499(1)**
F_8(2D)	0.97(6)	0.938(8)	0.993(5)	0.602(10)	**1(1)**	0.807(9)	0.078(12)	**1(1)**	0.405(11)	0.962(7)	**1(1)**	**1(1)**
F_9(2D)	0.703(5)	0.717(4)	**0.857(1)**	0.347(11)	0.667(8)	0.433(10)	0.093(12)	0.81(3)	0.67(7)	0.68(6)	0.667(8)	0.827(2)
F_{10}(2D)	0.6(5)	0.688(3)	**0.795(1)**	0.23(10)	0(12)	0.283(9)	0.065(11)	0.418(7)	0.49(6)	0.33(8)	0.74(2)	0.603(4)
F_{11}(2D)	0.653(7)	0.657(6)	**0.697(1)**	0.35(11)	0.667(2)	0.423(10)	0.05(12)	0.603(9)	0.63(8)	0.663(5)	0.667(2)	0.667(2)
F_{11}(3D)	0.207(11)	0.547(8)	0.657(6)	0.27(9)	**0.667(1)**	0.213(10)	0.01(12)	0.66(5)	0.657(6)	**0.667(1)**	**0.667(1)**	**0.667(1)**
F_{12}(3D)	0.093(11)	0.23(8)	0.42(3)	0.1(10)	0.523(2)	0.133(9)	0.04(12)	0.263(6)	0.343(5)	0.26(7)	**0.627(1)**	0.373(4)
F_{11}(5D)	0(12)	0.043(10)	0.163(7)	0.15(8)	**0.667(1)**	0.11(9)	0.013(11)	0.643(5)	0.53(6)	0.663(4)	**0.667(1)**	**0.667(1)**
F_{12}(5D)	0(12)	0.015(8)	0.215(6)	0.01(10)	0.003(11)	0.08(7)	0.013(9)	0.243(5)	0.263(3)	0.25(4)	**0.403(1)**	0.348(2)
F_{11}(10D)	0(10)	0(10)	0.033(8)	0(10)	0.187(6)	0.033(8)	0.163(7)	0.31(3)	0.19(5)	0.293(4)	0.633(2)	**0.667(1)**
F_{12}(10D)	0(9)	0(9)	0.015(8)	0(9)	0(9)	0.06(6)	0.07(5)	0.098(3)	0.093(4)	0.113(2)	0.018(7)	**0.248(1)**
F_{12}(20D)	0(7)	0(7)	0(7)	0(7)	0(7)	0(7)	0.125(4)	0.24(3)	0.125(4)	0.245(2)	0.005(6)	**0.365(1)**
total rank	131	117	71	173	93	161	202	79	118	88	46	29



(a) (b) (c) (d)

Fig. 6. Convergence speeds of the multimodal algorithms on composite multimodal functions. (a) F_{11}(3D) (b) F_{11}(5D) (c) F_{11}(10D) (d) F_{12}(20D).

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

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

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

[10] 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.

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

[12] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Computat.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.

[13] X. Li, "A multimodal particle swarm optimizer based on fitness Euclidean-distance ration," in *Proc. Genet. Evol. Computat. Conf*., 2007, pp. 78–85.

[14] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multi-modal optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, Jun. 2013.

[15] M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Finding multiple global optima exploiting differential evolution's niching capability," in *IEEE Symposium on Differential Evolution (SDE)*, April 2011, pp. 1–8.

[16] M. G. Epitropakis, X. Li, and E. K. Burke, "A dynamic archive niching differential evolution algorithm for multimodal optimization," in *Proc. Congr. Evol. Comput.*, Cancun, Mexico, 2013, pp. 79–86.

[17] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization,"*IEEE Trans.Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.

[18] 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.

[19] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Doctoral dissertation, Comput. Commun. Sci., Univ. Michigan, Ann Arbor, MI, 1975.

[20] S. Mahfoud, "Niching methods for genetic algorithms," Doctoral dissertation, Comput. Sci., Univ. Illinois, Urbana, IL, 1995.

[21] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," in *Proc. Conf. Genet. Evol. Comput*., 2005, pp. 873–880.

[22] J. Holland, *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press, 1975.

[23] K. Pearson, "The problem of the random walk," *Nature*, vol. 72, pp.294-203, 1905.

[24] J. Zhang and A. Sanderson, "JADE: Adaptive differential evolution with optional external archive,"*IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[25] R. Ursem, "Multinational evolutionary algorithms," in Proc. *IEEE Congr. Evol. Computat.*, 1999, pp. 1633–1640.

[26] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for cec'2013 special session and competition on niching methods for multimodal function optimization," Evolutionary Computation and Machine Learning Group, RMIT University, Melbourne, Australia, Tech. Rep., 2013.