

A Dynamic Archive Niching Differential Evolution Algorithm for Multimodal Optimization

Michael G. Epitropakis*, Xiaodong Li[†], and Edmund K. Burke*

*Department of Computing Science and Mathematics, School of Natural Sciences, University of Stirling,
Stirling FK9 4LA, Scotland, email: mge@cs.stir.ac.uk, e.k.burke@stir.ac.uk

[†]School of Computer Science and IT, RMIT University,
VIC 3001, Melbourne, Australia, email: xiaodong.li@rmit.edu.au

Abstract—Highly multimodal landscapes with multiple local/global optima represent common characteristics in real-world applications. Many niching algorithms have been proposed in the literature which aim to search such landscapes in an attempt to locate as many global optima as possible. However, to locate and maintain a large number of global solutions, these algorithms are substantially influenced by their parameter values, such as a large population size. Here, we propose a new niching Differential Evolution algorithm that attempts to overcome the population size influence and produce good performance almost independently of its population size. To this end, we incorporate two mechanisms into the algorithm: a control parameter adaptation technique and an external dynamic archive along with a reinitialization mechanism. The first mechanism is designed to efficiently adapt the control parameters of the algorithm, whilst the second one is responsible for enabling the algorithm to investigate unexplored regions of the search space and simultaneously keep the best solutions found by the algorithm. The proposed approach is compared with two Differential Evolution variants on a recently proposed benchmark suite. Empirical results indicate that the proposed niching algorithm is competitive and very promising. It exhibits a robust and stable behavior, whilst the incorporation of the dynamic archive seems to tackle the population size influence effectively. Moreover, it alleviates the problem of having to fine-tune the population size parameter in a niching algorithm.

I. INTRODUCTION

Several algorithms/methodologies have been proposed in the literature of Evolutionary Computation (EC) that attempt to optimize problems with complex and highly multimodal landscapes. Some of them are capable of finding numerous global or/and local optima and, in several cases, the goal is to accurately detect as many as possible. As such, various Evolutionary Computation algorithms have been extended to address multimodal problems, namely the “niching” algorithms [1], [2]. Niching algorithms usually impose a restrictive reproduction technique in order to maintain the diversity of their populations and converge in parallel to multiple solutions. Over the last 20 years or so, various niching techniques have been proposed in the EC literature. Characteristic examples include: fitness sharing [2], [3], crowding [1], [2], fitness sharing [2], [3], clustering [4], clearing [5], speciation [6], restricted tournament selection [7], stretching and deflation [8], parallel approaches [9], and specialized operators [10], [11].

In the current work, given a multimodal function the objective is to locate as many global optima as possible. A successful niching algorithm should accomplish two major

objectives: It should accurately locate and maintain good solutions (local or global optima). To accomplish these two objectives, existing algorithms incorporate various niching techniques to induce the required *niching effect*. Most of them introduce new parameters that have to be carefully fine-tuned, such as the niche or species radius [12]–[14], and the crowding factor [15]. This is because a careless selection of their values may result in poor performance. In addition, one common parameter that plays a crucial role in the performance of a niching algorithm is its population size. In an unknown multimodal problem, where prior knowledge about the number of global optima or their characteristics is not available, a user has to fine tune those parameters. In the case of the population size, the user should either select a very large population size or carefully fine-tune it, which might not always be an effective approach especially for problems with very expensive function evaluations.

In this study, we aim to effectively tackle the aforementioned dependence on the manual selection of the population size value and we are attempting to design an algorithm that is not influenced by the size of its population. As such, we are going to employ a recently proposed Niching Differential Evolution algorithm [10], namely the DE/nrand/1 algorithm, which has the advantage of being able to reliably locate many global optima, without introducing an extra niching parameter. Taking DE/nrand/1 as a baseline model, we propose a parameter independent algorithm by incorporating two additional mechanisms into its structure: a well known control parameter adaptation technique [16] and an external dynamic archive along with a reinitialization mechanism [17]. The adaptive control parameter technique will alleviate the problem of having to fine-tune the standard control parameters required by Differential Evolution, i.e. the mutation and recombination factor [18]–[20]. On the other hand, the dynamic archive along with the reinitialization mechanism will be responsible for keeping the best potential solutions found by the algorithm and it will simultaneously re-initialize some individuals to allow the algorithm to search unexplored regions of the problem space. As a result, the algorithm is able to continue its search for additional good global solutions, without being bound by an initial small population size, resulting in a niching algorithm with its performance being almost independent from the population size parameter, i.e. small populations should be

sufficient to tackle complex multimodal problems.

The remainder of the paper is organized as follows: The motivation behind the development of the proposed algorithm, as well as some related work, is briefly described in Section II, whilst its main characteristics and implementation details are presented and discussed in Section III. In Section IV, we evaluate the proposed algorithm and discuss its outcomes based on extensive experimental results. The paper ends in Section V with a brief discussion of the contribution of this work.

II. RELATED WORK

In the EC literature, there are several Differential Evolution (DE) variants that utilize some niching technique in an attempt to tackle multimodal problems with a high number of global and/or local optima. Thomsen adopted the classical deterministic crowding and the fitness sharing technique, and produced two well known algorithms, the Crowding DE (CDE) and the Sharing DE [15]. Their performance was evaluated on several multimodal problems, where the Crowding DE outperformed the Sharing DE in the majority of the considered problems [15]. The selection pressure, imposed in CDE, is determined by the *crowding factor* which is the key parameter for its performance. In turn, Li in [12] employed the concept of speciation to DE's structure, resulting in the widely used Species-based DE (SDE) [12], [13]. In principle, SDE locates simultaneously multiple global optima by adaptively splitting its population into multiple species and evolving them through the DE algorithm. The species are determined by the *species radius* parameter, which is a user-specified and problem dependent parameter. As such, a careless selection of its value may result in deterioration of the algorithm's performance.

Moreover, DE with local selection [21] designs a mutation strategy with two main components: a local and a global mutation rule. Throughout the evolutionary process, the two rules are probabilistically selected by a fixed and pre-specified probability. As expected, the global mutation rule is responsible for investigating unexplored regions of the search space, whilst the local mutation rule contributes towards its exploitative ability. The algorithm has been further hybridized with several mechanisms such as the crowding technique and a specialized multi-start local search procedure [22]. A parallel implementation of DE has been proposed by Zaharie to address multimodal problems [9]. In this approach, Zaharie uses the classic "island model" to locate multiple global optima simultaneously. Recent niching DE variants include ensembles of niching techniques such as the restricted tournament selection [23], the incorporation of crowding, fitness sharing and speciation in DE with the concept of neighborhood mutations [24]. Apart from the DE algorithm, various methodologies have been introduced in the EC literature which aim to optimize multimodal problems. Representative examples include the topological species conservation approach [14], Particle Swarm Optimizers [25]–[27], evolutionary strategies [28], multi-objective algorithms [29], and artificial immune systems [30], [31].

Furthermore, a new family of mutation operators has been recently proposed in [10], namely the DE/nrand family, which is able to efficiently address multimodal problems. Their key component is that, to induce the required *niching effect*, they simply incorporate information about the spatial neighborhoods of the population into the mutation operators of the algorithm. Thus, the resulting algorithms can locate and maintain several global optima in their populations, without introducing any additional parameters. Although the aforementioned mutation operators have stable performance through the evolution process, the DE/nrand family considers nearest neighbor calculations for each individual, which incurs an increase in the computational cost. Nevertheless, this can be handled by employing index-based neighborhoods with various topology structures [11].

It is evident that the majority of the aforementioned niching algorithms introduce one or more additional parameters, apart from the control parameter of the used algorithm, e.g. the mutation and recombination factors in the case of DE. In turn, to the best of our knowledge, there has been no attempt to tackle the population size parameter effectively in the niching DE algorithms literature. However, a Particle Swarm Optimization algorithm has been proposed which adopts a dynamic archive mechanism to alleviate the problem of having to tune the swarm size parameter [17]. Initial experimental results exhibit competitive and robust performance, resulting in a very promising approach [17].

Inspired by the above findings, we take as a baseline model the DE/nrand family and propose a parameter-free niching algorithm by incorporating two additional mechanisms within its structure: a well known control parameter adaptation technique from one of the best performing DE variants [16] and the aforementioned external dynamic archive along with a reinitialization mechanism [17]. The proposed approach, its main characteristics and its implementation details are briefly described in the following section.

III. NICHING DIFFERENTIAL EVOLUTION WITH DYNAMIC ARCHIVE

Motivated by the simple structure of the DE/nrand family of mutation strategies [10], the proposed approach takes as a baseline model the DE/nrand/1 algorithm. DE/nrand/1 follows the basic structure of the classic Differential Evolution algorithm [18] (i.e. utilization of the binomial crossover and the elitist selection operator) and interacts only with the mutation strategy. In general, information about the real (spatial) neighborhoods of the population is incorporated into DE's mutation strategies. As such, each individual is evolved by using (as a base vector) its nearest neighbor individual and random vector differences in an attempt to keep the individual within the vicinity of an optimum and simultaneously to explore effectively the search space.

More specifically, given a population of NP individuals, at each generation g , the mutation strategy will evolve each individual x_g^i , $i = 1, 2, \dots, NP$, to generate the mutant

individual v_{g+1}^i according to the following equation:

$$v_{g+1}^i = x_g^{NN_i} + F(x_g^{r_1} - x_g^{r_2}), \quad (1)$$

where $x_g^{NN_i}$ is the spatial nearest neighbor of the current individual x_g^i , $F > 0$ is the *mutation* or *scaling factor*, and r_1, r_2 are randomly chosen indexes ($r_1, r_2 \in \{1, 2, \dots, NP\} \setminus \{i\}$).

DE has two main control parameters that affect the search procedure: the *mutation* and *recombination* factor. The main role of the *mutation* factor is to control the impact of the vector's differences, which mainly influences the convergence rate of the algorithm [19]. The *recombination* is mainly responsible for the diversity of the population. Improper selection of their values may lead to very poor performance and destroy the dynamics of the algorithm. Several techniques have been proposed in the literature to select, adapt or self adapt their values [16], [32]. Here, to appropriately select the values of these parameters, we adopt the technique proposed in the context of the JADE algorithm, which is one of the most successful DE techniques in the literature [32].

Briefly, JADE associates each individual, x_g^i , with a mutation (F_i) and a recombination (CR_i). At each generation, both parameters are drawn from specific distributions with different parameters ($CR_i \sim N(\mu_{CR}, 0.1)$, $F_i \sim \text{Cauchy}(\mu_F, 0.1)$). After the main evolution steps, the parameter of the distributions are updated according to the following equations:

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F), \quad (2)$$

$$\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR}), \quad (3)$$

where μ_{CR} and μ_F are the corresponding mean values, c is a constant with $c \in [0, 1]$ usually $c = 0.1$, $\text{mean}_A(\cdot)$ denotes the mean value of a vector, S_{CR} denotes a vector with the successful recombination probabilities of the current generation and $\text{mean}_L(\cdot)$ is the Lehmer mean, which can be calculated as $\text{mean}_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}$. Further details about the parameter adaptation technique can be found in [16].

Moreover, the main objective of the proposed algorithm is to alleviate the problem of having to tune the population size and allow the algorithm to maintain good performance regardless of the population size value. To accomplish this goal, we incorporate a dynamic archive proposed in [17] to record good solutions found by the algorithm along with a re-initialization procedure to continue searching for additional good solutions in unexplored regions of the search space. Specifically, when a new individual x_{g+1}^i is selected for the next generation it will be determined whether or not to be inserted into the dynamic archive. If the new individual is qualified for insertion then it is kept in the archive. If the solution is already in the archive (Algorithm 1, lines 20-22) then this solution is re-initialized within the bounds of the problem at hand, in an attempt to search for unexplored regions. The algorithmic scheme of the proposed algorithm (dADE/nrand/1) and the dynamic archive are briefly illustrated in Algorithm 1 and 2 respectively.

In general, the archive has the ability to record the ‘‘good’’ solutions found during evolution. A potential solution p is added in the archive only if the archive does not contain any

Algorithm 1

The algorithmic scheme for dADE/nrand/1.

```

1: Initialize population with  $NP$  individuals,  $g = 0$ 
2: while termination criteria not satisfied do
3:   Calculate the  $R$  value based on Eq. (4).
4:   for  $i = 1$  to  $NP$  do
5:     Update parameters  $F$  and  $CR$  based on Eqs. (2),(3)
6:     Select individuals  $x_g^{r_1}, x_g^{r_2}$ , and  $x_g^{NN_i}$ 
7:     /*Mutate  $x_g^i$  and generate the mutant vector  $v_g^{i,*}$ */
8:     for  $j = 1$  to  $D$  do
9:        $v_g^{i,j} = x_g^{NN_i} + F(x_g^{r_1,j} - x_g^{r_2,j})$ ,
10:    end for
11:     $j_{\text{rand}} =$  a uniformly distributed random integer  $\in \{1, 2, \dots, D\}$ 
12:    /*Recombine the mutant vector  $v_g^{i,*}$ */
13:    for  $j = 1$  to  $D$  do
14:       $u_g^{i,j} = \begin{cases} v_g^{i,j}, & \text{if } (\text{rand}_{i,j}(0,1) \leq CR \text{ or } j = j_{\text{rand}}), \\ x_g^{i,j}, & \text{otherwise,} \end{cases}$ 
15:    end for
16:    /*Select the final vector  $x_{g+1}^{i,*}$ */
17:    if  $f(u_g^i) < f(x_g^i)$  then
18:       $x_{g+1}^i = u_g^i$ 
19:      Insert  $x_{g+1}^i$  to the Dynamic Archive based on Algorithm 2.
20:      if  $x_{g+1}^i$  is already in Archive (found == TRUE) then
21:        Re-initialize individual  $x_{g+1}^i$ 
22:      end if
23:    else
24:       $x_{g+1}^i = x_g^i$ 
25:    end if
26:  end for
27:   $g = g + 1$ 
28: end while

```

Algorithm 2

The algorithmic scheme for building a dynamic archive.

Input: $p \in \mathbb{R}^D$ a potential solution, $R \in \mathbb{R}$ the niche radius, and ε an accuracy level or acceptance threshold

Output: S a list of solutions that are kept in the dynamic archive

```

1: found  $\leftarrow$  FALSE; update  $\leftarrow$  FALSE;
2: if  $S = \emptyset$  then
3:    $S \leftarrow S \cup \{p\}$ ;  $\delta \leftarrow f(p)$ ;
4: else
5:   if  $f(p) > \delta$  then
6:      $\delta \leftarrow f(p)$ ; update  $\leftarrow$  TRUE;
7:   end if
8:   if update or  $|f(p) - \delta| < \varepsilon$  then
9:     for each  $s \in S$  do
10:      if  $\|p - s\| \leq R$  then
11:        if  $f(p) > f(s)$  then
12:           $s \leftarrow p$ ;
13:          found  $\leftarrow$  TRUE;
14:          break;
15:        else
16:          found  $\leftarrow$  TRUE;
17:          break;
18:        end if
19:      end if
20:    end for
21:    if not found then
22:       $S \leftarrow S \cup \{p\}$ ;
23:    end if
24:  end if
25: end if

```

better or similar solutions. We consider that two solutions are similar, if their Euclidean distance is smaller than the identification radius R (see below in Eq. (4)). As depicted in Algorithm 2, each potential solution is compared with the saved solutions in the archive. If a potential solution p has

either a better fitness value than the best fitness value found so far (i.e. the δ threshold), or a relatively good fitness value compared with the best solution in the archive ($|f(p) - \delta| < \varepsilon$, where ε is an acceptable level of accuracy), then it will be compared with the saved solutions in the archive. p is added in the archive, only if the archive does not contain any similar solutions.

The key idea behind the identification radius R is to decide whether a newly found solution is in the vicinity of a globally optimum solution or not. In the first stages of evolution, it is easy to identify the large basins of attraction. Thus, large radius values are sufficient. However, as the evolution process continues, discovering niches located on smaller basins of attraction becomes a major challenge. Thus, as suggested in [33] the identification radius R can be adaptively chosen during evolution. As such, the radius R , is calculated as the minimum among the average distances of the individuals' positions at the current generation g . Specifically, R can be calculated according to the following equations:

$$R = \min\{r_1, r_2, \dots, r_g\}, \quad (4)$$

where r_g reflects the convergence degree of all individuals in the population, and can be defined as $r_g = (\sum_{i=1}^n \text{dist}_i)/n$ where $\text{dist}_i = \min\{\|x_g^i - x_g^j\| : \forall x_g^i, x_g^j \in \text{in the population} \setminus x_g^i \neq x_g^j\}$.

IV. EXPERIMENTAL RESULTS

In this section, we investigate the performance and the characteristics of the proposed approach, by providing comparative experimental results with other DE variants that are designed to tackle multimodal optimization problems. Their effectiveness is verified through a benchmark function set proposed recently in the "IEEE CEC'2013 Special Session and Competition on Niching Methods for Multimodal Optimization" [34]. The benchmark function set contains twenty instances of twelve multimodal functions with various characteristics. The first eight are well-known simple low-dimensional multimodal functions, whilst the remaining four are scalable composition functions of challenging multimodal functions. Most of the functions share common properties such as multiple local and global optima, deceptiveness, non-symmetric optima and non-separable optima. A detailed description of the benchmark function set and its characteristics can be found in the associated technical report [34].

To evaluate the efficiency of the proposed algorithm (dADE/nrand/1), we compare it with two DE variants: the well known Crowding Differential Evolution [15] and the recently proposed DE/nrand/1 algorithm [10]. Throughout this section, we adopt the experimental design and the parameter settings proposed in [34]. As such, for each experimental result presented in this section, we conduct 50 independent runs and report their average values. In terms of the control parameters for DE, we have used $F = 0.5$ and $CR = 0.9$ which is a common parameter setting in the literature [10], [19], [20], [34]. In addition, to make it easy to compare the approaches with the presented results in [34], for each experiment, we have

employed populations with $NP = 100$ individuals. It has to be noted that the aforementioned parameter setting is not fine-tuned on the considered benchmark set, and the parameters should not be considered to be optimal for the considered functions.

To measure if the algorithm at hand is able to locate accurately the desired number of global optima, we initially have to define a tolerance level in which a computed solution can be considered as a global optimum, entitled the *level of accuracy*, $\varepsilon \in (0, 1]$. Thereby, a computed solution is characterized as a global optimum if its distance from a known global optimum is less than the given *level of accuracy* ε . Usually, the applied distance corresponds to the Euclidean distance but it can be easily changed, based on the problem's characteristics. Afterwards, to determine if all global optima are found, we adopt the Algorithm 1 proposed in [34]. The performance of the implemented algorithms is compared using two well known measures, as suggested by the CEC'2013 competition [34], namely the *peak ratio* (PR) and the *success rate* (SR) [15].

Three sets of experiments have been conducted. Firstly, we evaluate the proposed algorithm on the recently proposed benchmark suite defined in [34]. In turn, we study the behavior of the dADE/nrand/1 algorithm in two different qualitative aspects: its convergence speed characteristics and its sensitivity to various population size values. To this end, the first objective is to study the performance of the proposed algorithm in terms of locating a large number of global optima through different levels of accuracy. Based on the experimental procedure proposed in [34], we incorporate five levels of accuracy, $\varepsilon \in \{10^{-1}, 10^{-2}, \dots, 10^{-5}\}$. It can be easily observed that, as the accuracy level increases, this task is getting more challenging, since the positions of the global optima should be computed more accurately. Generally speaking, the performance of niching algorithms tends to decrease, in terms of both PR and SR measures, as the accuracy level increases. Due to space limitations, we demonstrate experimental results for the proposed dADE/nrand/1 algorithm, and its ancestor DE/nrand/1, whilst we omit the experimental results for the Crowding DE algorithm, which (as presented in [34]) performs either similarly or worse, when compared with the DE/nrand/1 algorithm on most of the functions.

Tables I and II present extensive experimental results over all benchmark functions for the DE/nrand/1 and the proposed dADE/nrand/1 algorithm, respectively. Here, we mark in **bold-face** the algorithm that exhibits better performance in terms of either the PR or the SR measure. Based on the experimental results, in the majority of the considered functions, the proposed algorithm (dADE/nrand/1) exhibits either better or equally good performance as compared with DE/nrand/1, in terms of both PR and SR measures. Specifically, in the first five functions, both algorithms behave similarly, exhibiting best performance independently of the accuracy level. This behavior changes in the next five functions ($F_6(2D)$ – $F_7(3D)$), where dADE/nrand/1 algorithm clearly outperforms its ancestor DE/nrand/1 on both PR and SR measures. It

TABLE I
PEAK RATIOS AND SUCCESS RATES OF THE DE/NRAND/1/BIN
ALGORITHM.

Accuracy level ε	F_1 (1D)		F_2 (1D)		F_3 (1D)		F_4 (2D)		F_5 (2D)	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-03	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-04	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-05	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Accuracy level ε	F_6 (2D)		F_7 (2D)		F_8 (2D)		F_6 (3D)		F_7 (3D)	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	0.450	0.000	0.347	0.000	0.108	0.000	0.097	0.000	1.000	1.000
1.0E-02	0.438	0.000	0.346	0.000	0.105	0.000	0.095	0.000	1.000	1.000
1.0E-03	0.440	0.000	0.349	0.000	0.113	0.000	0.099	0.000	0.998	0.980
1.0E-04	0.434	0.000	0.337	0.000	0.112	0.000	0.095	0.000	1.000	1.000
1.0E-05	0.000	0.000	0.333	0.000	0.113	0.000	0.094	0.000	1.000	1.000
Accuracy level ε	F_9 (2D)		F_{10} (2D)		F_{11} (2D)		F_{11} (3D)		F_{12} (3D)	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	0.683	0.000	0.855	0.240	0.667	0.000	0.667	0.000	0.522	0.000
1.0E-02	0.673	0.000	0.837	0.220	0.667	0.000	0.667	0.000	0.535	0.000
1.0E-03	0.683	0.000	0.815	0.140	0.667	0.000	0.667	0.000	0.507	0.000
1.0E-04	0.673	0.000	0.815	0.160	0.667	0.000	0.667	0.000	0.502	0.000
1.0E-05	0.670	0.000	0.777	0.100	0.667	0.000	0.667	0.000	0.507	0.000
Accuracy level ε	F_{11} (5D)		F_{12} (5D)		F_{11} (10D)		F_{12} (10D)		F_{12} (20D)	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	0.677	0.000	0.345	0.000	0.403	0.000	0.227	0.000	0.130	0.000
1.0E-02	0.663	0.000	0.325	0.000	0.343	0.000	0.167	0.000	0.127	0.000
1.0E-03	0.663	0.000	0.295	0.000	0.323	0.000	0.152	0.000	0.130	0.000
1.0E-04	0.663	0.000	0.290	0.000	0.270	0.000	0.125	0.000	0.125	0.000
1.0E-05	0.657	0.000	0.287	0.000	0.250	0.000	0.127	0.000	0.123	0.000

TABLE II
PEAK RATIOS AND SUCCESS RATES OF THE PROPOSED
dADE/NRAND/1/BIN ALGORITHM.

Accuracy level ε	F_1 (1D)		F_2 (1D)		F_3 (1D)		F_4 (2D)		F_5 (2D)	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-02	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-03	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-04	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0E-05	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Accuracy level ε	F_6 (2D)		F_7 (2D)		F_8 (2D)		F_6 (3D)		F_7 (3D)	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	1.000	1.000	1.000	1.000	0.985	0.500	0.837	0.020	1.000	1.000
1.0E-02	1.000	1.000	0.962	0.240	0.978	0.380	0.595	0.000	1.000	1.000
1.0E-03	1.000	1.000	0.892	0.020	0.981	0.280	0.545	0.000	1.000	1.000
1.0E-04	0.984	0.780	0.823	0.000	0.967	0.140	0.431	0.000	1.000	1.000
1.0E-05	0.000	0.000	0.732	0.000	0.947	0.020	0.356	0.000	1.000	1.000
Accuracy level ε	F_9 (2D)		F_{10} (2D)		F_{11} (2D)		F_{11} (3D)		F_{12} (3D)	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	0.893	0.640	0.998	0.980	0.743	0.140	0.923	0.700	1.000	1.000
1.0E-02	0.667	0.000	0.887	0.440	0.667	0.000	0.667	0.000	0.620	0.000
1.0E-03	0.667	0.000	0.745	0.000	0.667	0.000	0.667	0.000	0.615	0.000
1.0E-04	0.667	0.000	0.740	0.000	0.667	0.000	0.667	0.000	0.627	0.000
1.0E-05	0.667	0.000	0.728	0.000	0.667	0.000	0.667	0.000	0.620	0.000
Accuracy level ε	F_{11} (5D)		F_{12} (5D)		F_{11} (10D)		F_{12} (10D)		F_{12} (20D)	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1.0E-01	0.873	0.540	0.938	0.760	0.683	0.080	0.420	0.000	0.030	0.000
1.0E-02	0.667	0.000	0.472	0.000	0.660	0.000	0.143	0.000	0.000	0.000
1.0E-03	0.667	0.000	0.417	0.000	0.630	0.000	0.063	0.000	0.002	0.000
1.0E-04	0.667	0.000	0.403	0.000	0.633	0.000	0.018	0.000	0.005	0.000
1.0E-05	0.667	0.000	0.410	0.000	0.627	0.000	0.000	0.000	0.000	0.000

is worth noting that the proposed algorithm exhibits great performance gains for the majority of the aforementioned functions for almost all accuracy levels. This is especially the case in functions $F_6(2D)$ – $F_6(3D)$ dADE/nrand/1 where it is able to successfully locate a greater number of global optima for several accuracy levels, whilst DE/nrand/1 is not. In turn, in the group of the last ten functions (Composition

functions), which are the most challenging functions of the function set, both algorithms exhibit a problem dependent performance. dADE/nrand/1 exhibits a superior performance for the first accuracy level in the majority of the functions in this group, whilst as the accuracy level increases both algorithms seem to behave differently. In three functions, both algorithms perform equally well ($F_9(2D)$, $F_{11}(2D)$, $F_{11}(3D)$) but they cannot manage to locate accurately all global optima. dADE/nrand/1 exhibits superior performance gains in three functions ($F_{11}(5D)$, $F_{12}(5D)$, $F_{11}(10D)$) comparing with DE/nrand/1, whilst DE/nrand/1 performs better in the last two functions ($F_{12}(10D)$ and $F_{12}(20D)$). In general, the usage of the archive seems to have substantial performance gains in the majority of the tested functions. Concluding, based on the aforementioned initial results, it seems that the proposed approach is very promising and can efficiently tackle very challenging multimodal problems.

A. Convergence speed characteristics

In this section, we investigate the convergence speed characteristics of all considered algorithms. Thus, for the algorithm at hand, convergence speed is determined by the number of function evaluations that the algorithm requires to locate all global optima, within a fixed level of accuracy, ε . Due to limited space, we compare only the DE/nrand/1 algorithm with the proposed dADE/nrand/1, since it was the best performing algorithm for the majority of the considered functions as presented in [34]. We evaluate their performance in two qualitative different accuracy levels, $\varepsilon \in \{1.0E - 01, 1.0E - 04\}$. The former corresponds to the easiest level of accuracy, whilst the latter corresponds to one of the most demanding cases considered in this work. Thus, in Table III we present the average performance of the algorithms (*Mean*) and its standard deviation (*St.D.*). We mark in **boldface** the algorithm that performs better, i.e. it has a lower mean number of function evaluations. In addition, we underline the instances where the corresponding algorithm performs significantly better. The statistically significant differences of the observed convergence speed performance have been verified by a two-sided Wilcoxon rank sum test at the 5% significance level.

It can be clearly observed, in Table III, that in the majority of the considered functions the dADE/nrand/1 produces faster convergence speed. The performance gains are more evident in the first accuracy level ($\varepsilon = 10^{-1}$), since it outperforms DE/nrand/1 in 13 out of 20 cases. Specifically, on the last 15 functions, the proposed approach exhibits better performance (lower mean number of function evaluations), indicating that it can accurately detect the basin of attractions of the considered functions within the given budget of function evaluations. In contrast, DE/nrand/1 performs significantly better in the first five low-dimensional functions. This is expected, since the proposed scheme incorporates a re-initialization procedure which may require a higher number of function evaluations in some cases. As such, in low dimensional or easily solvable functions, the aforementioned procedure might increase the required number of function evaluations in order to detect all

TABLE III
CONVERGENCE SPEEDS OF THE DE/nrand/1 AND THE PROPOSED dADE/nrand/1 ALGORITHM IN TWO ACCURACY LEVELS $\varepsilon \in \{1.0E-01, 1.0E-04\}$.

Accuracy level $\varepsilon = 1.0E-01$											
Algorithm	Function	$F_1(1D)$	$F_2(1D)$	$F_3(1D)$	$F_4(2D)$	$F_5(2D)$	$F_6(2D)$	$F_7(2D)$	$F_6(3D)$	$F_7(3D)$	$F_8(2D)$
DE/nrand/1	Mean	5768.0	200.0	200.0	2740.0	274.0	200000.0	200000.0	400000.0	400000.0	3162.0
	St.D.	1442.000	0.000	0.000	500.612	100.631	0.000	0.000	0.000	0.000	695.727
dADE/nrand/1	Mean	5922.1	221.0	203.4	3106.8	367.2	27458.5	2910.9	367281.6	396811.4	3391.8
	St.D.	1672.615	38.493	14.429	845.307	120.634	6903.683	617.711	42450.483	22547.090	652.852
Algorithm	Function	$F_9(2D)$	$F_{10}(2D)$	$F_{11}(2D)$	$F_{11}(3D)$	$F_{12}(3D)$	$F_{11}(5D)$	$F_{12}(5D)$	$F_{11}(10D)$	$F_{12}(10D)$	$F_{12}(20D)$
DE/nrand/1	Mean	200000.0	164480.0	200000.0	400000.0	400000.0	400000.0	400000.0	400000.0	400000.0	400000.0
	St.D.	0.000	64129.919	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
dADE/nrand/1	Mean	145455.6	114735.2	182184.9	219868.9	61965.4	292772.7	200502.8	392376.0	340214.0	400000.0
	St.D.	58240.354	21749.001	47527.121	154229.939	16890.382	133135.821	127782.460	33324.142	95903.583	0.000
Accuracy level $\varepsilon = 1.0E-04$											
Algorithm	Function	$F_1(1D)$	$F_2(1D)$	$F_3(1D)$	$F_4(2D)$	$F_5(2D)$	$F_6(2D)$	$F_7(2D)$	$F_6(3D)$	$F_7(3D)$	$F_8(2D)$
DE/nrand/1	Mean	22886.0	1552.0	1258.0	13610.0	3806.0	200000.0	200000.0	400000.0	400000.0	9858.0
	St.D.	2689.056	386.106	781.179	1399.453	618.890	0.000	0.000	0.000	0.000	833.015
dADE/nrand/1	Mean	20201.6	1800.8	1289.5	12703.2	3567.1	150328.0	200000.0	393666.8	400000.0	12903.7
	St.D.	2787.875	586.007	565.283	1668.362	651.549	35209.408	0.000	17665.255	0.000	2168.516
Algorithm	Function	$F_9(2D)$	$F_{10}(2D)$	$F_{11}(2D)$	$F_{11}(3D)$	$F_{12}(3D)$	$F_{11}(5D)$	$F_{12}(5D)$	$F_{11}(10D)$	$F_{12}(10D)$	$F_{12}(20D)$
DE/nrand/1	Mean	200000.0	181658.0	200000.0	400000.0	400000.0	400000.0	400000.0	400000.0	400000.0	400000.0
	St.D.	0.000	42543.630	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
dADE/nrand/1	Mean	200000.0	200000.0	200000.0	400000.0	400000.0	400000.0	400000.0	400000.0	400000.0	400000.0
	St.D.	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

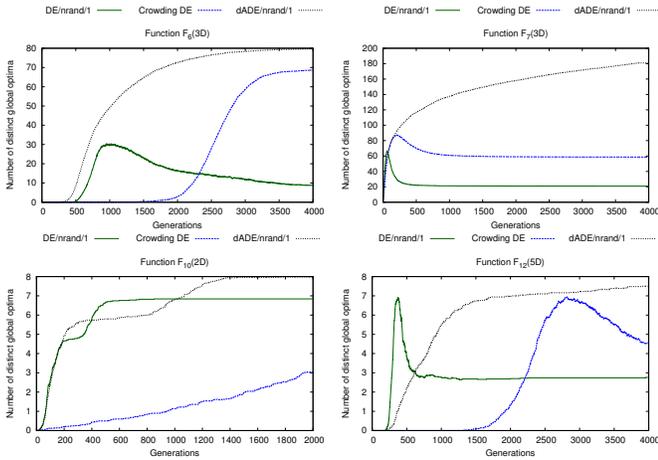


Fig. 1. Mean number of global optima located by the algorithms on functions $F_6(3D)$, $F_7(3D)$, $F_{10}(2D)$, and $F_{12}(5D)$ over 50 independent experiments (accuracy level $\varepsilon = 10^{-1}$).

the global optima. In the second and more challenging level of accuracy, it can be clearly observed that we cannot draw firm conclusions since both algorithms for the majority of the functions did not succeed in locating all global optima solutions. However, in the first 10 functions, dADE/nrand/1 has either better (in $F_1(1D)$, $F_4(2D)$, $F_5(2D)$, $F_6(2D)$, and $F_6(3D)$) or equally good ($F_3(1D)$, $F_7(2D)$, and $F_7(3D)$) convergence speed when comparing with DE/nrand/1.

Moreover, Figure 1 demonstrates the mean number of global optima found by the considered algorithms during the experiments, in the accuracy level of $\varepsilon = 10^{-1}$. It can be easily observed that the proposed approach exhibits a stable and robust behavior in all considered cases. As expected, the nature of the archive enriches the characteristics of the algorithm and helps maintain the located global optima during evolution. In contrast, on most functions both DE/nrand/1 and

Crowding DE are able to discover the basis of attraction of several global optima in the first stages of the evolution, but tend to lose them at later stages. For example, in functions $F_7(3D)$ and $F_{12}(5D)$, both algorithms detect a high number of global optima in the early stages of evolution, but cannot maintain them for a long period of time, e.g. for about 500 and 1000 generations in $F_7(3D)$ and $F_{12}(5D)$ respectively. Comparing DE/nrand/1 with the Crowding DE, it can be clearly observed that in most functions DE/nrand/1 detects many global optima faster, but it cannot maintain them in its population for a long time, whilst the Crowding DE tends to maintain them for longer periods of time. In summary, the proposed approach demonstrates a robust and stable behavior that does not demand a larger number of function evaluations as is the case with other algorithms.

B. Population size effect

The experimental results section ends by studying the ability of the considered algorithm to detect a large number of global optima against different population size values. As such, for a challenging accuracy level, $\varepsilon = 10^{-4}$, we select the two and three dimensional versions of Shubert ($F_6(2D)$, $F_6(3D)$) and Vincent ($F_7(2D)$, $F_7(3D)$), and for various population size values, $NP \in \{20, 30, \dots, 200\}$, we measure the number of the global optima found by the corresponding algorithm.

To this end, Figure 2 illustrates the mean number of global optima found by the considered algorithms for the different population size values. It can be clearly observed that dADE/nrand/1 performs well on all considered functions independently of the population size. The incorporation of the dynamic archive, along with the creation of new randomly generated solutions, enhances the ability of DE/nrand/1 algorithm to detect a large number of global optima and to keep them in its population until the end of the runs. Specifically, in the considered cases, dADE/nrand/1 is clearly able to detect a high number of global optima even when the population

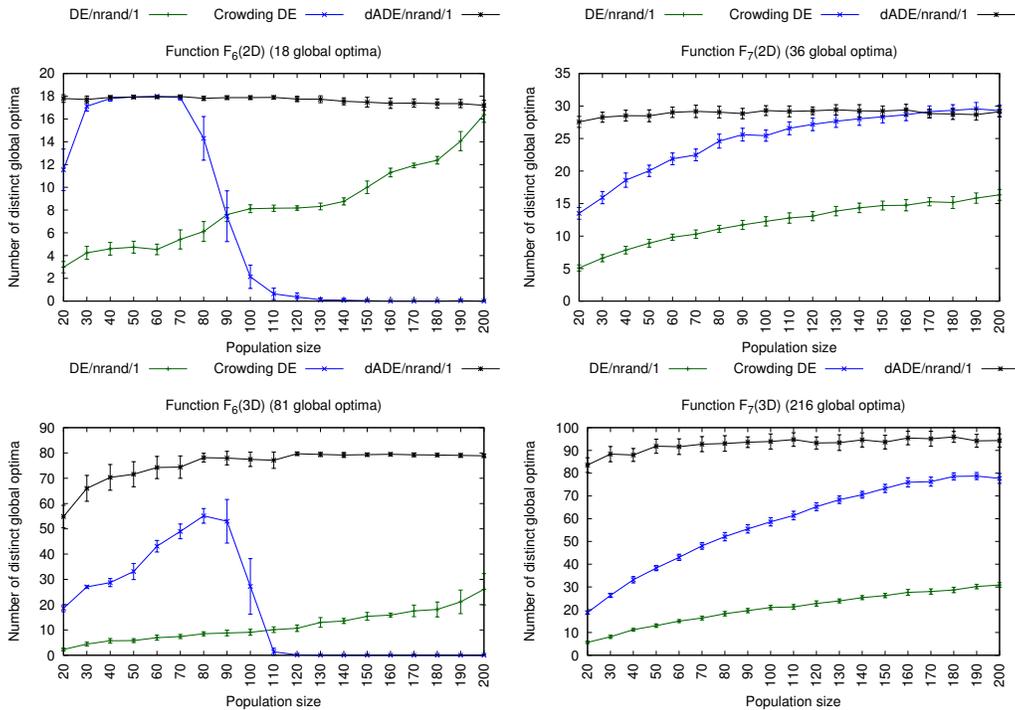


Fig. 2. Mean number of global optima found by the implemented algorithms against different population size values on the $F_6(2D)$, $F_7(2D)$, $F_6(3D)$ and $F_7(3D)$ functions ($\varepsilon = 10^{-4}$)

consists of only 20 individuals. It is worth noting that, for the three dimensional cases ($F_6(3D)$, $F_7(3D)$), dADE/nrand/1 can locate about 3 to 10 times more global optima solutions than the other approaches.

In contrast, the performance of both the Crowding DE and the DE/nrand/1 algorithm is inferior when compared with dADE/nrand/1, whilst they demonstrate a problem and algorithm dependent behavior. This is a very common phenomenon in the literature of multimodal optimization methodologies. The majority of the proposed methodologies are strongly influenced by the selected population size (often a large population size is essential to a greater number of global optima), whilst careless selections may result in substantial deterioration of the performance of the method at hand. Specifically, based on the algorithm’s performance, Figure 2 reveals two different trends. The DE/nrand/1 algorithm (in all four cases) and the Crowding DE (in both the 2D and 3D versions of F_7) exhibit stable performance trends against population size, where their performance increases as the population size increases. In addition, the Crowding DE (in both the 2D and 3D versions of F_6) exhibits high performance gains in a small range of population size values, i.e. when $NP \in \{30, 40, \dots, 80\}$ and $NP \in \{30, 40, \dots, 100\}$ for the 2D and 3D versions of the F_6 function respectively. Finally, we have observed that DE/nrand/1 performs worse against the Crowding DE because, with the fixed budget of function evaluations, it locates a high number of global optima but it cannot maintain all of them at the end of the execution.

To conclude, the incorporation of the dynamic archive seems

to tackle the population size influence effectively and it also alleviates the problem of having to fine-tune the population size parameter. Its incorporation within other state-of-the-art methodologies may produce similar effects.

V. CONCLUSIONS

It has been recognized that most of the niching algorithms introduce one or more parameters to the algorithm at hand. Additionally, the majority of the niching methodologies proposed in the literature are substantially influenced by the specified population size. In this work, we attempted to overcome the population size influence by proposing a parameter-free niching algorithm with robust performance regardless of the selection of the population size. To this end, we incorporate into a Niching Differential Evolution algorithm, namely DE/nrand/1, two mechanisms: a control parameter adaptation technique [16] and an external dynamic archive along with a reinitialization mechanism [17]. The first mechanism efficiently adapts the control parameters of DE, whilst the second mechanism is responsible for allowing the algorithm to explore uninvestigated regions of the search space and simultaneously to keep the best solutions found by the algorithm, independently of the population size.

Experimental verification on the recently proposed benchmark suite of the “IEEE CEC’2013 Special Session and Competition on Niching Methods for multimodal Optimization” [34] and comparisons with the two DE variants clearly indicate that the proposed algorithm produces competitive and very promising results. In the majority of the benchmark func-

tions, it locates and maintains a large number of global optima, resulting in a robust performance. Additionally, in most of the considered functions, the proposed approach is able to detect all global optima in fewer function evaluations when compared against the considered DE variants. Initial experimental studies indicate that the proposed algorithm is able to detect a very large number of global optima with minimal population size values, which makes it an appealing technique for multimodal problems with unknown characteristics.

Future work includes extensive investigation of the dynamics, the characteristics, the scalability and the complexity of the proposed algorithm on multimodal problems with higher dimensions.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments. E.K. Burke and M.G. Epitropakis would like to thank EPSRC for their support for this work through grant EP/J017515/1. X. Li was supported by an ARC Discovery Grant (No. DP120102205).

REFERENCES

- [1] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems." Ph.D. dissertation, University of Michigan, USA, 1975.
- [2] S. W. Mahfoud, "Nicheing methods for genetic algorithms," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1995, uMI Order No. GAX95-43663.
- [3] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, Hillsdale, NJ, USA, 1987, pp. 41–49.
- [4] D. K. Tasoulis, V. P. Plagianakos, and M. N. Vrahatis, "Clustering in evolutionary algorithms to efficiently compute simultaneously local and global minima," in *IEEE Congress on Evolutionary Computation*, vol. 2, 2005, pp. 1847–1854.
- [5] A. Pétrowski, "A clearing procedure as a niching method for genetic algorithms," in *IEEE International Conference on Evolutionary Computation*, 1996, pp. 798–803.
- [6] J. 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, 2002.
- [7] G. R. Harik, "Finding multimodal solutions using restricted tournament selection," in *Proceedings of the 6th International Conference on Genetic Algorithms*, San Francisco, CA, USA, 1995, pp. 24–31.
- [8] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis, "Objective function "stretching" to alleviate convergence to local minima," *Nonlinear Analysis*, vol. 47, no. 5, pp. 3419–3424, 2001.
- [9] D. Zaharie, "Extensions of differential evolution algorithms for multimodal optimization," in *6th International Symposium of Symbolic and Numeric Algorithms for Scientific Computing*, 2004, pp. 523–534.
- [10] M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Finding multiple global optima exploiting differential evolution's niching capability," in *2011 IEEE Symposium on Differential Evolution (SDE)*, April 2011, pp. 1–8.
- [11] —, "Multimodal optimization using niching differential evolution with index-based neighborhoods," in *IEEE Congress on Evolutionary Computation, 2012. CEC 2012. (IEEE World Congress on Computational Intelligence)*, Brisbane, Australia, June 2012, pp. 1–8.
- [12] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," in *Proceedings of the 2005 conference on genetic and evolutionary computation, GECCO 2005*, 2005, pp. 873–880.
- [13] J. Li, X. Li, and A. Wood, "Species based evolutionary algorithms for multimodal optimization: A brief review," in *2010 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, July 2010, pp. 1–8.
- [14] 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. PP, no. 99, p. 1, 2010.
- [15] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *IEEE Congress on Evolutionary Computation*, vol. 2, 2004, pp. 1382–1389.
- [16] J. Zhang and A. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.
- [17] Z. Zhai and X. Li, "A dynamic archive based niching particle swarm optimizer using a small population size," in *Proceedings of the Australian Computer Science Conference (ACSC 2011)*, M. Reynolds, Ed. Perth, Australia: ACM, 2011, pp. 1–7.
- [18] R. Storn and K. Price, "Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.
- [19] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.
- [20] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 99–119, 2011.
- [21] J. I. Rönkkönen and J. Lampinen, "An extended mutation concept for the local selection based differential evolution algorithm," in *Proceedings of the 9th annual conference on genetic and evolutionary computation*. London, England: ACM, 2007, pp. 689–696.
- [22] J. Rönkkönen, X. Li, V. Kyrki, and J. Lampinen, "A framework for generating tunable test functions for multimodal optimization," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, pp. 1–18, 2010.
- [23] B.-Y. Qu and P. Suganthan, "Novel multimodal problems and differential evolution with ensemble of restricted tournament selection," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–7.
- [24] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 601–614, Oct. 2012.
- [25] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 211–224, 2004.
- [26] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 150–169, 2010.
- [27] R. Brits, A. P. Engelbrecht, and F. V. D. Bergh, "A niching particle swarm optimizer," in *Proceedings of the Conference on Simulated Evolution And Learning*, 2002, pp. 692–696.
- [28] O. M. Shir, M. Emmerich, and T. Bäck, "Adaptive niche radii and niche shapes approaches for niching with the CMA-ES," *Evolutionary Computation*, vol. 18, no. 1, pp. 97–126, 2010.
- [29] K. Deb and A. Saha, "Finding multiple solutions for multimodal optimization problems using a multi-objective evolutionary approach," in *Proceedings of the 12th annual conference on genetic and evolutionary computation*. Portland, Oregon, USA: ACM, 2010, pp. 447–454.
- [30] F. Campelo, F. Guimaraes, H. Igarashi, J. Ramirez, and S. Noguchi, "A modified immune network algorithm for multimodal electromagnetic problems," *IEEE Transactions on Magnetics*, vol. 42, no. 4, pp. 1111–1114, 2006.
- [31] L. de Castro and J. Timmis, "An artificial immune network for multimodal function optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC '02*, vol. 1, 2002, pp. 699–704.
- [32] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2010.
- [33] S. Bird and X. Li, "Adaptively choosing niching parameters in a PSO," in *Proceedings of the 8th annual conference on Genetic and Evolutionary Computation*, ser. GECCO '06. New York, NY, USA: ACM, 2006, pp. 3–10.
- [34] 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. [Online]. Available: <http://goanna.cs.rmit.edu.au/~xiaodong/cec13-niching/competition/>