Finding Multiple Global Optima Exploiting Differential Evolution's Niching Capability

Michael G. Epitropakis
Computational Intelligence Laboratory,
Department of Mathematics,
University of Patras, Greece.
Email: mikeagn@math.upatras.gr

Vassilis P. Plagianakos
Department of Computer Science
and Biomedical Informatics,
University of Central Greece,
Greece. Email: vpp@ucg.gr

Michael N. Vrahatis
Computational Intelligence Laboratory,
Department of Mathematics,
University of Patras, Greece.
Email: vrahatis@math.upatras.gr

***Abstract*—Handling multimodal functions is a very important and challenging task in evolutionary computation community, since most of the real-world applications exhibit highly multimodal landscapes. Motivated by the dynamics and the proximity characteristics of Differential Evolution's mutation strategies tending to distribute the individuals of the population to the vicinity of the problem's minima, we introduce two new Differential Evolution mutation strategies. The new mutation strategies incorporate spatial information about the neighborhood of each potential solution and exhibit a niching formation, without incorporating any additional parameter. Experimental results on eight well known multimodal functions and comparisons with some state-of-the-art algorithms indicate that the proposed mutation strategies are competitive and very promising, since they are able to reliably locate and maintain many global optima throughout the evolution process.**

I. Introduction

Evolutionary Algorithms (EAs) are nature inspired stochastic search methods. They are based on the common concept of evolving a population of candidate solutions by simulating the main processes involved in the evolution of the genetic material of organism populations, such as natural selection and biological evolution. EAs can be characterized as global optimization algorithms and their population-based nature provides them with the ability to avoid being trapped in a local optimum; consequently a greater chance to find global optimal solutions exists. In many cases, EAs tend to converge to a single optimum solution, which drives the population to slowly lose its diversity through the evolution stages. Nevertheless, most hard real-world problems are considered to be highly multimodal problems. They are likely to have several global and/or local minima, and in many cases it is desirable to accurately locate as many as possible.

To this end, EAs have been extended through the concept of the niche formation. Niche formation is a common biological phenomenon [1]. A niche can be characterized as a subspace in the environment that can support different types of life. Niches could aid the differentiation of the species and thus maintain their diversity, by imposing reproduction restrictions. Many natural environments can lead to niche formation, for example, remote islands, high mountains and isolated valleys. Many well-known EAs, characterized as Niching methods, have been implemented to mimic the biological niche formation and take advantage of its characteristics [1]. Niching methods tend to maintain the diversity within their population and allow a parallel convergence into multiple solutions. Several niching techniques have been proposed and successfully applied to different EAs, namely, crowding [2], [3], fitness sharing [3], [4], clearing [5], clustering [6], stretching and deflation [7], [8], parallelization [9], restricted tournament selection [10], [11], and speciation [12].

In this paper, we study in particular the Differential Evolution (DE) algorithm proposed by Storn and Price [13]. This method has been applied in a plethora of optimization problems with high success [13]–[16]. Without loss of generality, we only consider minimization problems. In this case, the objective is, given a multimodal function, to locate as many global minimizers as possible. Several different DE variants incorporate the aforementioned niching techniques and attempt to handle multimodal problems. Thomsen extends DE with both a crowding and a fitness sharing technique, namely Crowding DE (CDE) and Sharing DE [17]. In the aforementioned work Thomsen showed that the CDE variant is a more promising approach, since outperforms the Sharing DE in all tested problems [17]. Generally, the crowding technique [2], [3] changes the selection procedure and allows a competition for limited resources among similar individuals. The similarity of individuals is calculated by a distance measure between their genotypes, i.e. Euclidean distance. Specifically, the technique compares an offspring with randomly chosen individuals of the current population and if the offspring is fitter, replaces the most similar one. Crowding technique generally maintains a better population diversity and therefore prevents premature convergence to an optimum.

Species-based DE (SDE) [18] incorporates the speciation concept to handle multimodal functions. SDE locates multiple global optima simultaneously through the adaptive formation of multiple species. Each species is evolved by its own DE process, which tries to successively improve itself. The algorithm will eventually be able to locate multiple global optima. Although SDE is computationally more efficient than the Crowding DE, it incorporates a user-specified and problem dependent parameter called *species radius*, which should be properly chosen. Furthermore, DE using local selection (DELS) [19] employs a new mutation strategy that divides the mutation operation into the local and the global mutation stages. With a pre-specified probability, it selects a different mutation strategy to perform either a global or a local mutation. The global mutation helps the algorithm to explore the search space, while the local mutation to efficiently search locally. DELS has been further hybridized with a multi-start gradient-based local search, as well as with the crowding technique [20]. In [9], an "island model" approach is incorporated to locate many global optima in parallel, while in [11] a DE extension with an ensemble of the restricted tournament selection (ERTS-DE) has been proposed. Finally, several other evolutionary computing methods have been proposed that attempt to deal with multimodal fitness landscapes [8], [21]–[25].

Recent studies on the dynamics and the proximity capabilities of the DE algorithm have shown that after a number of steps the candidate solutions tend to gather around optimizers of the objective function [6], [16], [26]. Exploiting the aforementioned behavior may enhance the performance of several DE variants [16]. In this respect, here we introduce two new mutation strategies that are able to efficiently handle multimodal functions. The new mutation strategies incorporate information regarding the neighborhood of each potential solution, which aids them to accurately locate and maintain many global optimizers, without the need of additional parameters. The proposed mutation strategies are evaluated on eight well-known and widely used multimodal problems, and their performance is compared against some state-of-the-art algorithms. Simulation results suggest that the proposed strategies are very promising and exhibit competitive behavior.

The rest of the paper is structured as follows: Section II briefly describes the behavior of different mutation operators, motivating us to introduce the proposed mutation strategies. A brief description of the Differential Evolution algorithm along with the new mutation strategies are presented in Section III. Next, the outcomes of an extensive experimental analysis are presented in Section IV and finally the paper concludes in Section V with a discussion and some pointers for future work.

II. Motivation

In recent works, the impact of the dynamics of different DE mutation strategies on the population of individuals has been investigated [6], [16], [26]. Experimental simulations indicate that DE mutation strategies tend to distribute the individuals of the population in the vicinity of the minima of the objective function. Exploitative strategies rapidly gather all the individuals to the basin of attraction of a single minimum, while explorative strategies tend to spread the individuals around many minima.

In [16], we showed that by exploiting the characteristics of the aforementioned tendency and incorporating them into the structure of the DE its performance is enhanced. To quantify the aforementioned tendency and to properly study the behavior of different DE mutation strategies we have incorporated a *clustering tendency* statistic, namely the H-measure [16], [27]. Clustering tendency is a well known concept in the cluster analysis literature that deals with the problem of determining the presence or absence of a clustering structure in a data set. Large values of the H-measure indicate the presence of a clustering structure in the dataset, while small values indicate the presence of regularly spaced points. A value around 0.5 indicates that the dataset contains randomly distributed vectors over the search space.

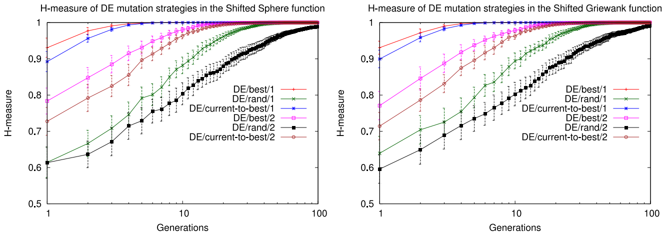

Fig. 1. H-measure of six classic DE mutation strategies on the Shifted Sphere and on the Shifted Griewank

Thereby, to investigate the behavior of six well known DE mutation strategies [16], we have selected two qualitatively different functions; the Shifted Sphere and the Shifted Griewank function. The former is a simple unimodal, while the latter is a highly multimodal function. Figure 1, illustrates mean values of the H-measure calculated at each generation for the six DE mutation strategies. The mean values have been obtained from 100 independent simulations for the 30-dimensional versions of the Shifted Sphere and Shifted Griewank functions. Error bars around the mean depict the standard deviation of the H-measure. All mutation strategies exhibit large H-measure values within the first 100 generations, indicating a strong clustering structure. Notice that DE/best/1 is the most exploitative strategy, while the least exploitatives seem to be DE/rand/1 and DE/rand/2.

Inspired by these findings, we introduce two new mutation strategies that incorporate population's neighborhood information and are able to efficiently locate and maintain multiple global optima of a multimodal function.

III. Differential Evolution mutation strategies for handling multimodal functions

The DE algorithm [13] is a stochastic parallel direct search method, which utilizes concepts borrowed from the broad class of Evolutionary Algorithms (EAs). More specifically, DE is a population–based stochastic algorithm that exploits a population of NP potential solutions, *individuals*, to effectively probe the search space. Firstly, DE randomly initializes the population in the D–dimensional optimization domain by utilizing a uniform probability distribution. Individuals evolve over successive iterations to explore the search space and locate the minima of the objective function. Throughout the evolution process, the user–defined population size, NP, is fixed. At each iteration, called *generation*, new vectors are derived by the combination of randomly chosen vectors from the current population. This operation in our context can be referred to as *mutation*, while the outcoming vectors as *mutant individuals*. Several mutation strategies have been proposed in the DE literature, The most common and widely used can be found in [13]–[16]. To continue, each mutant individual is then mixed with another, predetermined, vector – the *target* vector – through an operation called *recombination* or *crossover*. This operation yields the so–called *trial* vector. The most well known and widely used variants of DE utilize two main crossover schemes; the *exponential* and the *binomial* or *uniform crossover* [13]–[15]. Finally, the trial vector undergoes the *selection* operator, according to which it is accepted as a member of the population of the next generation only if it yields a reduction in the value of the objective function f relative to that of the target vector. Otherwise, target vector is retained in the next generation. The search operators efficiently shuffle information among the individuals, enabling the search for an optimum to focus on the most promising regions of the solution space. A more comprehensive description of the DE can be found in [13]–[16].

In this work, we take advantage from the dynamics and the clustering tendency of the classic DE/rand/1 and DE/rand/2 mutation strategies, and incorporate into their mutation schemes local information from the current population. To efficiently locate and maintain global optima, we evolve each individual by applying as a base vector its nearest neighbor individual, in an attempt to maintain the individual to the vicinity of an optimum and simultaneously to effectively explore the search space by incorporating random vectors into the differences of the mutation scheme. More specifically, for each individual x_g^i, $i = 1, 2, \ldots, NP$, where g denotes the current generation, the mutant individual v_{g+1}^i can be generated according to one of the following new proposed equations:

1) **"DE/nrand/1"**

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

2) **"DE/nrand/2"**

$$v_{g+1}^i = x_g^{NN_i} + F(x_g^{r_1} - x_g^{r_2}) + F(x_g^{r_3} - x_g^{r_4}), \qquad (2)$$

where $x_g^{NN_i}$ is the nearest neighbor of the current individual x_g^i, $r_1, r_2, r_3, r_4 \in \{1, 2, \ldots, NP\} \setminus \{i\}$ are random integers mutually different and not equal to the running index i, and $F > 0$ is a real parameter, called *mutation or scaling factor*. The mutation factor F, controls the amplification of the difference between two individuals, and is used to prevent the risk of stagnation of the search process. The computational complexity of the proposed mutation strategies is determined by the computational burden of the nearest neighbor computations. In the worst case the proposed strategies exhibit a similar complexity with the Crowding DE method [17]. The main goal of this work is to study if the incorporation of the nearest neighbor concept in the mutation operator can produce a "niching effect" without employing any additional parameter. Thus, in a future work, we intend to further study its complexity and efficiently determined the nearest neighbors by utilizing properly computational geometry methods [28]. Finally, the rest of the DE steps remain the same as the classic DE/rand/1/bin scheme [13], i.e. we employ the binomial crossover operator and the simple elitist selection operator.

IV. Experimental Results

In this section, we perform an experimental evaluation of the proposed approaches and compare their performance with several algorithms that can handle multimodal problems. To verify the effectiveness of the proposed approaches we employ eight well known and widely used multimodal benchmark functions having different characteristics, such as multiple evenly and unevenly spaced global optima, multiple global optima in the presence of multiple local optima and deceptiveness [20]. Table I shows the benchmark functions and some of their characteristics. More specifically, functions F_1 and F_2 have a low number of irregularly spaced global minima and no local minima. Function F_3 has 18 global minima and a high number (742) of local minima, with the global minima situated in two groups of nine regularly spaced minima in a three times three square shape. Function F_5 has 6^D global minima without local minima and is partially irregular, resembling a Bezier stretched function of the cosine family, with the differences between minima to increase along the value of y. Function F_6 contains 5^D evenly spaced global minima and does not have any local minima. Similarly, function F_7 has the same number of minima, but the distances between each global minimum decrease towards the origin. Finally, function F_8 is the modified version of the well–known Rastrigin function, having 4 evenly spaced global minima and 96 local minima.

To demonstrate the efficiency of the proposed approaches, we compare them with five methods, i.e. the two classic DE/rand/1/bin and DE/rand/2/bin algorithms and three methods that have been designed to handle multimodal problems, namely the FERPSO [22], the Crowding DE [17], and the DELS [19]. Throughout this section, all the reported results are averaged over 100 independent simulations. For each simulation and each method we have initialized the populations using a uniform random number distribution with the same random seeds. Furthermore, all methods have been implemented with the default parameters settings as have been proposed in the literature. Regarding the DE control parameters, the common setting of $F = 0.5$ and $CR = 0.9$ were used for all DE variants [14], [15]. Regarding the FERPSO parameters, the proposed setting of $\phi_{\max} = 4.1$ and $\chi = 0.729$ has been utilized [22]. The population size has been kept fixed to $NP = 100$ individuals and for each simulation, a budget of $\max NFEs = 10^5$ function evaluations has been employed.

To verify the effectiveness and the ability of the proposed approaches to accurately locate the global minima, we first specify a *level of accuracy*, $\varepsilon \in (0, 1]$. The *level of accuracy* depicts the tolerance level of a computed solution to considered as a global optimum. Specifically, if the Euclidean distance of a computed solution to a known global optimum is less than the pre-specified level of accuracy ε, then we consider the solution to be a global optimum. Furthermore, since in the current benchmark functions the number and the location of optima to be found is known *a priori*, we can use it as a performance metric. Thus, based on the aforementioned metric, to compare the performance of the implemented

TABLE I
EIGHT MULTIMODAL BENCHMARK FUNCTIONS [20]

Function	Mathematical formula	Optimization box	D	# global minima	# local minima
Branin	$F_1(\vec{y}) = \left(y_2 - \frac{5.1}{4\pi^2}y_1^2 + \frac{5}{\pi}y_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(y_1) + 10$	$y_1 \in [-5, 10], y_2 \in [0, 15]$	2	3	–
Himmelblau	$F_2(\vec{y}) = (y_1^2 + y_2 - 11)^2 + (y_1 + y_2^2 - 7)^2$	$\vec{y} \in [-6, 6]^2$	2	4	–
Shubert	$F_3(\vec{y}) = \sum_{i=1}^{5} i\cos((i+1)y_1 + i) \cdot \sum_{i=1}^{5} i\cos((i+1)y_2 + i)$	$\vec{y} \in [-10, 10]^2$	2	18	742
Six-hump camel back	$F_4(\vec{y}) = \left(4 - 2.1y_1^2 + \frac{y_1^4}{3}\right)y_1^2 + y_1y_2 + (-4 - 4y_2^2)y_2^2$	$y_1 \in [-1.9, 1.9]$, $y_2 \in [-1.1, 1.1]$	2	2	4
Vincent	$F_5(\vec{y}) = -\frac{1}{D}\sum_{i=1}^{D}\sin(10 \cdot \log(y_i))$	$\vec{y} \in [0.25, 10]^D$	2	6^D	–
Deb 1	$F_6(\vec{y}) = -\frac{1}{D}\sum_{i=1}^{D}\sin^6(5\pi y_i)$	$\vec{y} \in [0, 1]^D$	2	5^D	–
Deb 3	$F_7(\vec{y}) = -\frac{1}{D}\sum_{i=1}^{D}\sin^6(5\pi(y_i^{\frac{3}{4}} - 0.05)$	$\vec{y} \in [0, 1]^D$	2	5^D	–
Modified Rastrigin	$F_8(\vec{y}) = 20 + \sum_{i=1}^{2}\left(y_i^2 + 10\cos(2\pi y_i)\right)$	$\vec{y} \in [-5.12, 5.12]^2$	2	4	96

TABLE II
SUCCESS RATIO AND PEAK RATIO MEASURES FOR THE MULTIMODAL FUNCTIONS F_1–F_4

Function F_1	FERPSO		DE/rand/1		DE/rand/2		DE/nrand/1		DE/nrand/2		Crowding DE		DELS	
Accuracy level ε	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
1.0e-03	0.00	0.103	0.00	0.333	0.00	0.570	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-04	0.00	0.113	0.00	0.333	0.00	0.577	**1.00**	**1.000**	0.99	0.997	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-05	0.00	0.107	0.00	0.333	0.00	0.547	**1.00**	**1.000**	0.99	0.997	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-06	0.00	0.113	0.00	0.333	0.00	0.587	**1.00**	**1.000**	**1.00**	**1.000**	0.91	0.970	**1.00**	**1.000**
1.0e-07	0.00	0.150	0.00	0.333	0.00	0.563	**1.00**	**1.000**	**1.00**	**1.000**	0.12	0.493	**1.00**	**1.000**
1.0e-08	0.00	0.100	0.00	0.333	0.00	0.570	**1.00**	**1.000**	**1.00**	**1.000**	0.00	0.067	**1.00**	**1.000**
1.0e-09	0.00	0.133	0.00	0.333	0.00	0.570	**1.00**	**1.000**	0.99	0.997	0.00	0.010	**1.00**	**1.000**
Function F_2	FERPSO		DE/rand/1		DE/rand/2		DE/nrand/1		DE/nrand/2		Crowding DE		DELS	
Accuracy level ε	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
1.0e-03	0.00	0.333	0.00	0.250	0.00	0.250	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-04	0.00	0.333	0.00	0.250	0.00	0.250	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-05	0.00	0.347	0.00	0.250	0.00	0.250	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-06	0.00	0.320	0.00	0.250	0.00	0.250	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-07	0.01	0.317	0.00	0.250	0.00	0.250	**1.00**	**1.000**	**1.00**	**1.000**	0.96	0.990	**1.00**	**1.000**
1.0e-08	0.00	0.292	0.00	0.250	0.00	0.250	**1.00**	**1.000**	**1.00**	**1.000**	0.11	0.522	**1.00**	**1.000**
1.0e-09	0.00	0.335	0.00	0.250	0.00	0.250	**1.00**	**1.000**	**1.00**	**1.000**	0.00	0.083	**1.00**	**1.000**
Function F_3	FERPSO		DE/rand/1		DE/rand/2		DE/nrand/1		DE/nrand/2		Crowding DE		DELS	
Accuracy level ε	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
1.0e-03	0.00	0.465	0.00	0.118	0.47	0.918	0.01	0.743	**0.92**	**0.996**	0.00	0.047	0.00	0.674
1.0e-04	0.00	0.437	0.00	0.112	0.45	0.872	0.01	0.717	**0.89**	**0.994**	0.00	0.003	0.00	0.148
1.0e-05	0.00	0.419	0.00	0.112	0.53	0.923	0.00	0.736	**0.95**	**0.997**	0.00	0.000	0.00	0.015
1.0e-06	0.00	0.411	0.00	0.118	0.51	0.858	0.01	0.727	**0.97**	**0.998**	0.00	0.000	0.00	0.003
1.0e-07	0.00	0.375	0.00	0.114	0.49	0.902	0.00	0.748	**0.97**	**0.998**	0.00	0.000	0.00	0.000
1.0e-08	0.00	0.356	0.00	0.106	0.42	0.854	0.01	0.729	**0.92**	**0.996**	0.00	0.000	0.00	0.000
1.0e-09	0.00	0.322	0.00	0.114	0.44	0.873	0.02	0.751	**0.91**	**0.995**	0.00	0.000	0.00	0.000
Function F_4	FERPSO		DE/rand/1		DE/rand/2		DE/nrand/1		DE/nrand/2		Crowding DE		DELS	
Accuracy level ε	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
1.0e-03	0.01	0.095	0.00	0.500	0.89	0.945	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-04	0.00	0.020	0.00	0.500	0.91	0.955	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-05	0.00	0.005	0.00	0.500	0.90	0.950	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-06	0.00	0.005	0.00	0.500	0.89	0.945	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-07	0.00	0.000	0.00	0.500	0.91	0.955	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-08	0.00	0.005	0.00	0.500	0.94	0.970	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-09	0.00	0.005	0.00	0.500	0.89	0.945	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**

algorithms, we adopt the following two measures: the *peak ratio* and the *success rate* [17]. Strictly speaking, for a pre-specified budget of function evaluations ($\max NFEs$) and an accuracy level ε, the *peak ratio* (PR) measures the percentage of global optima (i.e. peaks) located over the total number of known global optima. Thus, for one simulation the *peak ratio* can be defined as: PR $= \frac{\text{number of peaks found}}{\text{number of total peaks}}$. It should be noticed that the PR reported values are average values over 100 independent simulations. Additionally, the *success rate* (SR) measures the percentage of simulations in which *all* global optima have been successfully located.

The first task that we want to tackle is to find out if the implemented methods can accurately locate the global minima in different levels of accuracy. Thereby, Table II, Table III as well as Figure 2 exhibit extensive experimental results of all algorithms over all benchmark functions considered in this work. We evaluate the algorithms for seven different accuracy levels, namely $\varepsilon \in \{10^{-3}, 10^{-4}, \ldots, 10^{-9}\}$. This is a very challenging task, since when the accuracy level decreases the accuracy of the computed global minima increases. As it is

TABLE III

SUCCESS RATIO AND PEAK RATIO MEASURES FOR THE MULTIMODAL FUNCTIONS F_5–F_8

Function F_5	FERPSO		DE/rand/1		DE/rand/2		DE/nrand/1		DE/nrand/2		Crowding DE		DELS	
Accuracy level ε	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
1.0e-03	0.00	0.161	0.00	0.028	0.00	0.028	0.00	0.401	0.00	0.391	0.00	**0.718**	0.00	0.493
1.0e-04	0.00	0.155	0.00	0.028	0.00	0.028	0.00	0.400	0.00	0.389	0.00	**0.707**	0.00	0.486
1.0e-05	0.00	0.136	0.00	0.028	0.00	0.028	0.00	0.394	0.00	0.374	0.00	**0.621**	0.00	0.474
1.0e-06	0.00	0.131	0.00	0.028	0.00	0.028	0.00	0.371	0.00	0.351	0.00	0.342	0.00	**0.437**
1.0e-07	0.00	0.121	0.00	0.028	0.00	0.028	0.00	0.367	0.00	0.294	0.00	0.127	0.00	**0.402**
1.0e-08	0.00	0.107	0.00	0.028	0.00	0.028	0.00	**0.362**	0.00	0.213	0.00	0.016	0.00	0.350
1.0e-09	0.00	0.097	0.00	0.028	0.00	0.028	0.00	**0.339**	0.00	0.100	0.00	0.003	0.00	0.268
Function F_6	FERPSO		DE/rand/1		DE/rand/2		DE/nrand/1		DE/nrand/2		Crowding DE		DELS	
Accuracy level ε	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
1.0e-03	0.00	0.396	0.08	0.886	0.14	0.921	0.68	0.985	0.62	0.983	**1.00**	**1.000**	0.61	0.982
1.0e-04	0.00	0.389	0.04	0.885	0.14	0.917	0.67	0.984	0.62	0.981	**1.00**	**1.000**	0.61	0.980
1.0e-05	0.00	0.352	0.09	0.868	0.11	0.919	0.59	0.978	0.68	0.986	**1.00**	**1.000**	0.40	0.967
1.0e-06	0.00	0.313	0.11	0.870	0.16	0.923	0.54	0.976	0.64	0.983	**1.00**	**1.000**	0.01	0.602
1.0e-07	0.00	0.301	0.09	0.888	0.19	0.934	0.57	0.978	0.63	0.983	**1.00**	**1.000**	0.00	0.122
1.0e-08	0.00	0.270	0.07	0.878	0.18	0.930	0.69	0.985	0.64	0.982	**1.00**	**1.000**	0.00	0.016
1.0e-09	0.00	0.242	0.12	0.890	0.13	0.923	0.66	0.984	0.62	0.982	**0.99**	**1.000**	0.00	0.000
Function F_7	FERPSO		DE/rand/1		DE/rand/2		DE/nrand/1		DE/nrand/2		Crowding DE		DELS	
Accuracy level ε	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
1.0e-03	0.00	0.333	0.00	0.040	0.00	0.200	0.00	0.816	0.02	0.868	**1.00**	**1.000**	0.29	0.958
1.0e-04	0.00	0.305	0.00	0.040	0.00	0.212	0.00	0.796	0.01	0.831	**0.99**	**1.000**	0.20	0.952
1.0e-05	0.00	0.277	0.00	0.040	0.00	0.211	0.00	0.779	0.00	0.614	**0.09**	0.864	0.03	**0.881**
1.0e-06	0.00	0.254	0.00	0.040	0.00	0.200	0.00	**0.747**	0.00	0.204	0.00	0.236	0.00	0.483
1.0e-07	0.00	0.224	0.00	0.040	0.00	0.177	0.00	**0.696**	0.00	0.029	0.00	0.028	0.00	0.085
1.0e-08	0.00	0.197	0.00	0.040	0.00	0.199	0.00	**0.651**	0.00	0.003	0.00	0.002	0.00	0.010
1.0e-09	0.00	0.162	0.00	0.040	0.00	0.182	0.00	**0.561**	0.00	0.000	0.00	0.000	0.00	0.000
Function F_8	FERPSO		DE/rand/1		DE/rand/2		DE/nrand/1		DE/nrand/2		Crowding DE		DELS	
Accuracy level ε	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
1.0e-03	0.01	0.398	0.12	0.447	0.90	0.950	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-04	0.01	0.357	0.03	0.373	0.90	0.950	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-05	0.00	0.350	0.12	0.440	0.91	0.955	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-06	0.01	0.355	0.11	0.438	0.90	0.952	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-07	0.01	0.355	0.05	0.393	0.94	0.970	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-08	0.00	0.363	0.01	0.373	0.94	0.970	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**
1.0e-09	0.01	0.343	0.09	0.417	0.86	0.935	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**	**1.00**	**1.000**

expected, both the SR and the PR vary with respect to different levels of accuracy and, in general, the algorithms tend to produce smaller values of both peak ratio and success rates as the accuracy levels decrease.

In more detail, FERPSO exhibits poor performance on this benchmark set with the considered parameter settings. It can locate a low number of minima in almost all the considered functions, irrespective of the accuracy level. We speculate that FERPSO needs a higher swarm size to accurately locate all global minima on the given budget of function evaluations. Although the classic mutation strategies DE/rand/1 and DE/rand/2 have not be designed to tackle multimodal functions, in many cases, they exhibit good SR and PR values (e.g. functions F_3, F_4, F_6, and F_8). More specifically, DE/rand/1 always finds one global minimum, but does not maintain individuals on many global minima. On the other hand, DE/rand/2 exhibits a better "niching effect" on the population, resulting in good performance (e.g. functions F_3, F_4, F_6, and F_8). Crowding DE is one of the most promising approaches implemented in this work. It exhibits a very good performance (e.g. functions F_4, F_6, and F_8), but when the accuracy level decreases it can not maintain the population on many global minima (e.g. functions $F_1 - F_3, F_5$, and F_7). Thus, as depicted in Tables II and III, for the aforementioned functions, both SR and PR measures decrease with the level accuracy. A similar behavior can be observed for the DELS method, i.e. although it performs very good in functions F_1, F_2, F_4 and F_8, in the most challenging functions with many global minima (e.g. functions F_3, F_5, F_6, and F_7) its performance slowly decreases as the accuracy level decrease.

Finally, both the proposed DE/nrand/1 and DE/nrand/2 algorithms, exhibit good performance, in terms of SR and PR measures, for the majority of the considered functions (e.g. functions $F_1, F_2, F_3, F_4, F_6, F_7$, and F_8). Generally, they exhibit either equal or better performance when compared against the other utilized methods, especially when the accuracy level decreases (e.g. functions F_1, F_2, F_3, F_4, and F_8). More specifically, in most difficult problems (e.g. functions F_3, F_5, F_6, and F_7) DE/nrand/1 locates either all or a high number of global minima, irrespectively of the accuracy level. In the cases where the performance of Crowding DE and DELS is not good, DE/nrand/1 performs better in terms of PR and SR, e.g. in F_3, and the high accuracy levels of F_5 and F_7. DE/nrand/2 performs similarly in most of the functions and in the challenging F_3 function exhibits the best performance in comparison with all the other methods. Comparing the DE/nrand/1 with the DE/nrand/2 approach, the performance of the first approach is most robust in functions F_5 and F_7.

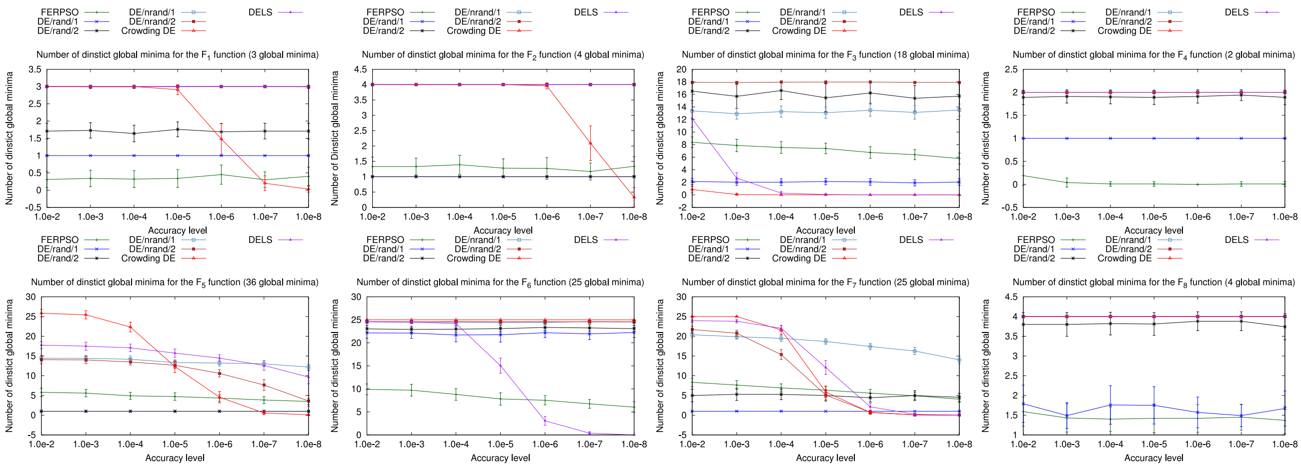

Fig. 2. Number of distinct global minima found by the implemented methods versus different levels of accuracy ε.

DE/nrand/1 can maintain a high number of global minima as the accuracy level decreases, while the performance of DE/nrand/2 tends to slowly decrease. Based on the aforementioned observations we believe that both proposed approaches exhibit a great potential for accurately locating many global minima and thus successfully tackling multimodal functions.

A. Convergence speed

In this section, we present experimental results for all utilized methods in terms of convergence speed. To measure the convergence speed of a method at a pre-specified level of accuracy, we calculate the required number of function evaluations in which it can accurately locate all global minima of the problem at hand. More specifically, for each problem and each algorithm we have conducted 100 simulations ($\varepsilon = 10^{-4}, NP = 100$, and $\max NFE = 10^5$). Table IV exhibits the mean number of function evaluations (Mean), its standard deviation (St.D.), the SR measure and the mean value of the PR measure over the F_1, F_2, F_4 and F_8 functions.

One can clearly observe that the algorithms DE/nrand/1 and DE/nrand/2 exhibit better performance (lower mean values in terms of function evaluations), with the DE/nrand/1 to be the best performing algorithm in all four cases considered here. On the other hand, in most of the cases FERPSO, DE/rand/1 and DE/rand/2, did not succeed to reach 100% success rate. DELS comes third in terms of mean number of function evaluations, closely following the performance of the DE/nrand/2 algorithm. Crowding DE, in three out of four functions (e.g. functions F_1, F_2, and F_4), requires the highest number of function evaluations to locate all minima.

TABLE IV
CONVERGENCE SPEED (ACCURACY LEVEL $\varepsilon = 10^{-4}$)

Function F_1	Mean	St.D.	SR	Mean PR
FERPSO	N/A	N/A	0.00	0.113
DE/rand/1	N/A	N/A	0.00	0.333
DE/rand/2	N/A	N/A	0.00	0.577
DE/nrand/1	**6982.00**	989.37	1.00	1.000
DE/nrand/2	10920.20	1380.99	0.99	0.997
Crowding DE	40137.00	5951.48	1.00	1.000
DELS	13361.00	1625.48	1.00	1.000
Function F_2	Mean	St.D.	SR	Mean PR
FERPSO	N/A	N/A	0.00	0.333
DE/rand/1	N/A	N/A	0.00	0.250
DE/rand/2	N/A	N/A	0.00	0.250
DE/nrand/1	**13504.00**	1521.96	1.00	1.000
DE/nrand/2	25400.00	2817.91	1.00	1.000
Crowding DE	48691.00	4498.40	1.00	1.000
DELS	22122.00	2345.02	1.00	1.000
Function F_4	Mean	St.D.	SR	Mean PR
FERPSO	N/A	N/A	0.00	0.020
DE/rand/1	N/A	N/A	0.00	0.500
DE/rand/2	2979.12	426.48	0.91	0.955
DE/nrand/1	**4151.00**	915.24	1.00	1.000
DE/nrand/2	5109.00	942.28	1.00	1.000
Crowding DE	14793.00	3393.50	1.00	1.000
DELS	6030.00	1205.33	1.00	1.000
Function F_8	Mean	St.D.	SR	Mean PR
FERPSO	7800.00	N/A	0.01	0.357
DE/rand/1	6400.00	700.00	0.03	0.373
DE/rand/2	8961.11	873.02	0.90	0.950
DE/nrand/1	**8222.00**	910.72	1.00	1.000
DE/nrand/2	11037.00	1021.45	1.00	1.000
Crowding DE	11995.00	1453.90	1.00	1.000
DELS	14975.00	1328.43	1.00	1.000

Additionally, in Figure 3, we illustrate the behavior of the methods during the simulations. More specifically, we perform 100 independent simulations and for a given accuracy level ($\varepsilon = 10^{-4}$), we record the number of global minima the method can locate throughout the simulation. It is obvious that, in most of the cases, the proposed DE/nrand/1 and DE/nrand/2 algorithms efficiently and accurately locate a high number of global minima and maintain them until the end of the simulation. DELS and Crowding DE exhibit similar behavior, but they tend to locate the global minima slowly. On the other hand, in many cases FERPSO, DE/rand/1, DE/rand/2 although they exhibit a good niching effect and succeed to locate several minima, they can not maintain them until the end of the simulation. This behavior tends to be more visible as the accuracy level decreases and can be captured by the PR measure. In Tables II and III, we observe that in the most challenging functions (e.g. functions F_3, F_5, and F_7) the proposed DE/nrand/1 approach locates and maintains many

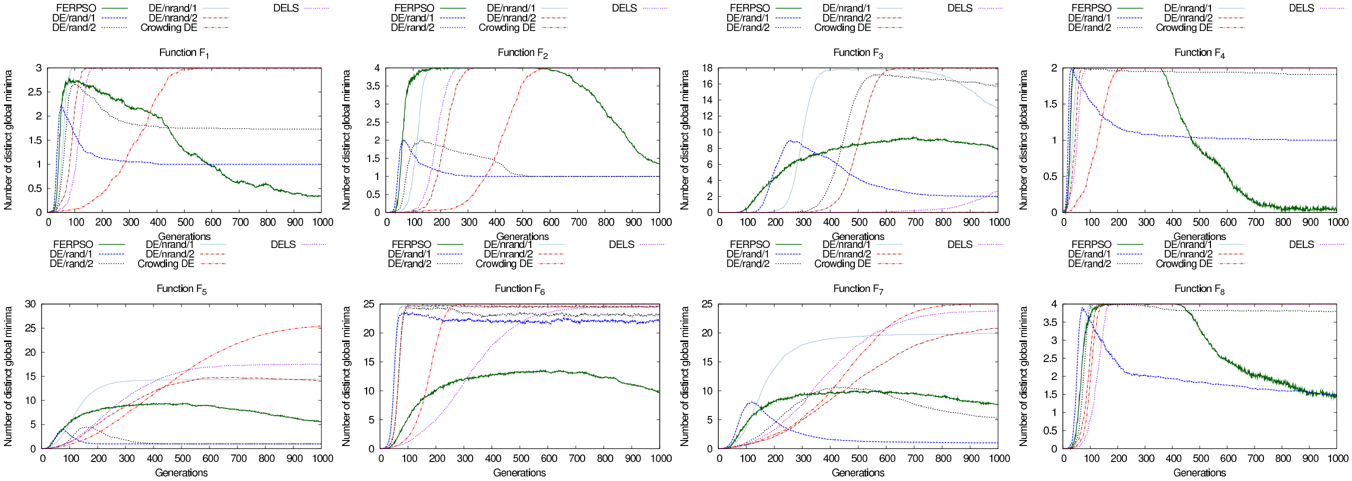

Fig. 3. Mean number of global minima found during 100 independent simulations of all methods over F_1–F_8 ($\varepsilon = 10^{-4}$)

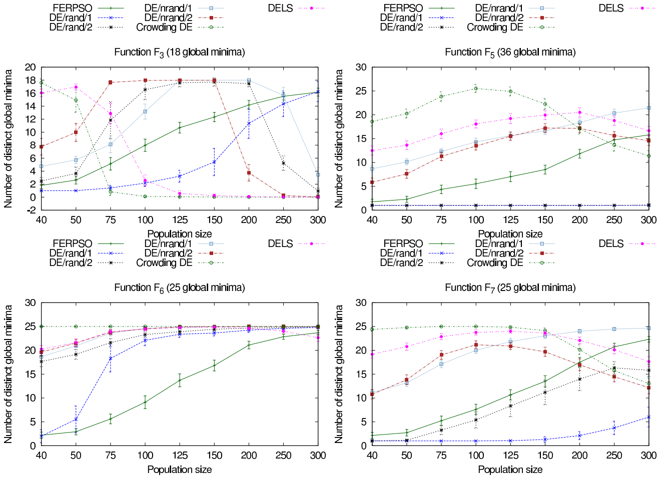

Fig. 4. Population size effect on the F_3, F_5, F_6 and F_7 functions

global minima as the accuracy level decreases, while Crowding DE and DELS do not (DE/nrand/1 exhibits higher values of the PR measure).

B. Population size effect

In this section, we study the effect of the population size of all implemented methods on the functions having many global minima (e.g. functions F_3, F_5, F_6, and F_7). For several different population sizes, we measure their performance by calculating the number of global minima found at the accuracy level $\varepsilon = 10^{-4}$ and within a budget of $\max NFEs = 10^5$ function evaluations. Figure 4 illustrates the performance of the methods as the population size increases from 40 to 300 individuals. Generally, in each function there exist population size values where most of the algorithms can locate a high number of global minima. As expected, this behavior depends on the structure of the problem at hand and the characteristics of the applied method. In function F_6, as the population size increase, almost all algorithms can locate more global minima. In functions F_5 and F_7, as the population increases DE/nrand/1, DE/nrand/2, FERPSO, Crowding DE and DELS manage to locate an increased number of global minima. Nevertheless, when the population size reaches 150–200 individuals the performance of DE/nrand/2, Crowding DE and DELS tends to rapidly decrease. Finally, in function F_3, Crowding DE and DELS can locate many minima with population sizes ranging from 40 to 50 individuals. When the population size increases their performance decrease rapidly, while their performance becomes marginal when the population size is larger than 100 individuals. On the contrary, DE/nrand/2, DE/nrand/1 and DE/rand/2 produce the best performance when the population size is between 100 and 150 individuals, and then as the population increases their performance decreases. Finally, DE/rand/1 and FERPSO exhibit a stable increasing performance as the population size increases, but in general their performance is worse than that of the other methods.

The most promising approaches in all considered functions are DE/nrand/1, DE/nrand/2, Crowding DE and DELS. The proposed DE/nrand/1 and DE/nrand/2 algorithms exhibit either better or equal performance in comparison with the other methods. DE/nrand/1 exhibits a more robust performance as the population size increases and in functions F_5 and F_7 outperforms the other methods, when the population size is from 250 or 300 individuals. DE/rand/2 can accurately locate many global minima, but when the population increases its performance tends to decrease. Crowding DE exhibited the best performance in F_5 and F_6 functions. Crowding DE and DELS exhibited good performance in some of the considered test functions (e.g. functions F_5 and F_6), but when the population size increases the performance of both methods decrease. Additionally, in the F_3 function, they exhibited a marginal performance for the majority of the population size values. FERPSO did not succeed to locate many global optima in the four functions considered in this section, but as discussed in [22] FERPSO may be capable of finding more global optima when a larger population size is employed. Thus, as one can observe it exhibits its best results when the population size is equal to 300 individuals.

Finally, DE/rand/1 in general, locates a small number of global optima, but it is interesting to observe that in functions F_3 and F_6 as the population increases the algorithm exhibits good niching performance. DE/rand/2 exhibits a better niching effect than DE/rand/1, while as the population size increases DE/rand/2 is capable to compute an increased number of global minima.

V. Conclusions

It has been recognized that throughout the evolutionary process of the Differential Evolution (DE) algorithm a clustering structure of the population of individuals can arise. In this study, we take advantage of the aforementioned behavior and attempt to improve DE's ability to accurately locate and maintain many global optima. To this end, we introduce two new mutation strategies that are based to the classic DE/rand/1 and DE/rand/2 strategies and incorporate into their schemes spatial information of the population. To evolve an individual, we apply as a base vector its nearest neighbor. Thus, the individuals effectively explore their neighborhoods.

Experimental results on eight well known multimodal functions with different characteristics and comparisons against two classic DE mutation strategies as well as three state-of-the-art algorithms, demonstrate that the proposed mutation strategies are competitive and very promising. Specifically, they can accurately locate many global optima and maintain them though the evolution process. In the four most challenging functions, they exhibit high peak ratio values and in most cases outperform the other algorithms. In terms of convergence speed, they can accurately locate all global minima in less function evaluations than the other algorithms. Finally, experiments regarding the effect of the population size show that, in most of the cases, the behavior of the proposed algorithms is robust.

In a future work, we will extensively study their performance and complexity on more multimodal function families, with higher dimensions and different characteristics.

Acknowledge

The authors would like to thank the European Social Fund (ESF), Operational Program for EPEDVM and particularly the Program Herakleitos II, for financially supporting this work.

References

[1] T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. Oxford University Press, 1997.

[2] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems." Ph.D. dissertation, University of Michigan, USA, 1975.

[3] S. W. Mahfoud, "Niching methods for genetic algorithms," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1995, uMI Order No. GAX95-43663.

[4] 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.

[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] D. Tasoulis, V. Plagianakos, and M. 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.

[7] 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.

[8] 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.

[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] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, 2010, 10.1109/TEVC.2010.2059031.

[16] 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*, 2010, 10.1109/TEVC.2010.2083670.

[17] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *IEEE Congress on Evolutionary Computation*, vol. 2, 2004, pp. 1382–1389.

[18] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," in *Proceedings of the 2005 conference on genetic and evolutionary computation*. ACM, 2005, pp. 873–880.

[19] 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.

[20] 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.

[21] 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.

[22] X. Li, "A multimodal particle swarm optimizer based on fitness euclidean-distance ratio," in *Proceedings of the 9th annual conference on genetic and evolutionary computation*. ACM, 2007, pp. 78–85.

[23] 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.

[24] 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.

[25] E. L. Yu and P. N. Suganthan, "Ensemble of niching algorithms," *Information Sciences*, vol. 180, pp. 2815–2833, August 2010.

[26] M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Balancing the exploration and exploitation capabilities of the differential evolution algorithm," in *IEEE Congress on Evolutionary Computation*, 2008, pp. 2686–2693.

[27] B. Hopkins and J. G. Skellam, "A new method for determining the type of distribution of plant individuals," *Annals of Botany*, vol. 18, no. 2, pp. 213–227, 1954.

[28] M. N. Vrahatis, B. Boutsinas, P. Alevizos, and G. Pavlides, "The new k-windows algorithm for improving the k-means clustering algorithm," *Journal of Complexity*, vol. 18, pp. 375–391, March 2002.