

Niche-Clearing-based Variable Mesh Optimization for Multimodal Problems

Ricardo Navarro
Department of Informatics
University of Holguín
Holguín, Cuba
rnavarro@facinf.uho.edu.cu

Rafael Falcon
Electrical Engineering & Computer Science
University of Ottawa
Ottawa, Canada
rfalcon@uottawa.ca

Rafael Bello
Department of Computer Science
Central University of Las Villas
Santa Clara, Cuba
rbello@uclv.edu.cu

Ajith Abraham
¹IT4Innovations, VSB-Technical University of Ostrava,
Czech Republic
²Machine Intelligence Research Labs (MIR Labs), USA
ajith.abraham@ieee.org

Abstract—The development of niching methods is a very active research area within multimodal optimization. It includes not only the creation of new methods, but the formalization of hybrid methodologies resulting from the application of basic niching techniques to global optimization metaheuristics. In this paper, we discuss some preliminary results of a recently proposed metaheuristic algorithm, Variable Mesh Optimization (VMO), in the context of multimodal problems. To overcome some of the encountered limitations, a revamped version called Niche-Clearing-based Variable Mesh Optimization (NC-VMO) is put forth. NC-VMO demonstrated its ability to optimize multimodal functions by using a niche clearing technique. Experimental results confirm that the proposed approach is also competitive with other niche-based optimization methods in the literature.

Keywords — *clearing procedure; niching methods; multimodal optimization; variable mesh optimization*

I. INTRODUCTION

Most real-coded evolutionary algorithms are devised for only locating a single global optimum, while ignoring other global (if any) and local optima as well. However, a variety of domains demand the search for multiple optima, may it be of local or global nature [40-42]. The last decades have witnessed an increasing interest paid to such problem by researchers in the evolutionary optimization community, to the extent that a new term (evolutionary multimodal optimization) has been coined [8, 22]. Inspired by ecosystems in nature, niching methods are a plausible attempt to approach multimodal optimization tasks.

All ecosystems have many different physical spaces (niches) with a finite amount of material resources to be shared among the individuals that belong to the same niche. Thus, the ecosystems allow not only the formation but also the maintenance of dissimilar species competing to survive. Niching methods encourage diversity by speciation, i.e. they split the population into distinct subpopulations (niches) occupying certain areas of the search space. In an optimization scenario, the fitness symbolizes the resources of

the niche and the species are defined as related individuals according to certain similarity metric, while every niche corresponds to an optimum of the fitness landscape [2]. Despite the large number of niche-based algorithms, there are still many drawbacks regarding such methods. For instance, most niching techniques are unable to solve a multimodal problem of a relatively large size or with a fairly large number of optima. In addition, drastic limitations on their computational complexity still persist.

Therefore, this is an active research area concerning both how to improve the performance of niching methods and how to hybridize them with global optimization metaheuristics so as to benefit from their synergy in order to deal with multimodality. As an evidence of the latter, there are many examples reported in the literature including niching not only in Genetic Algorithms (GA) [7, 12, 13, 21, 30], but in Particle Swarm Optimization [16, 26], Evolutionary Strategies [24] and so on.

Variable Mesh Optimization (VMO) [6] is a recently proposed metaheuristic for global optimization. It yielded competitive results when compared with outstanding state-of-the-art schemes in continuous optimization. VMO features three search operators, one aimed at global exploration and two for local optima exploitation. Thus, VMO seems to be a promising multimodal problem solver.

In this work, the following scientific contributions are made: (1) we shed light on empirical results regarding VMO's performance in multimodal optimization and (2) we put forward an improved VMO version termed "Niche-Clearing-based Variable Mesh Optimization" (NC-VMO), thus showcasing the benefits of coupling the canonical VMO formulation with a well-known niche formation procedure proposed by Pétrowski [4]. Besides the creation of a competitive VMO-based model, as far as we are concerned this is the first time niching methods are amalgamated with VMO. Empirical results confirm that this is a plausible mechanism to approximate multimodal functions.

The remainder of this paper is structured as follows. A short literature review is provided in Section II. Section III

elaborates on basic aspects of the VMO metaheuristic whereas the niche formation procedure is briefly described in Section IV. The hybrid niching-VMO scheme is presented in Section V. A time complexity analysis for both methods is made in Section VI. The experimental framework is detailed in Section VII while Section VIII discusses VMO and NC-VMO's performance on multimodal problems. Finally, conclusions and future work are outlined.

II. BRIEFLY REVIEWING NICHING METHODS

Literature on nature-inspired optimization algorithms includes a large variety of niching techniques. Cavicchio's pre-selection operator [12] is one of the first niching techniques. It was generalized in crowding by De Jong [21]. Mahfoud made subsequent modifications to crowding so as to reduce replacement errors, restore selection pressure and remove the parameter called "crowding factor" (CF), thus resulting in the deterministic crowding [30]. By means of a deterministic tournament, crowding favors higher-fitness individuals over lower-fitness ones. Hence, it leads to a loss of niches in a tournament between global and local niches. To avoid such deterministic nature, a technique named "probabilistic crowding" was introduced in [23], which uses a probabilistic replacement rule. In a later approach, the multi-niche crowding genetic algorithm (MNC GA) [33], both selection and replacement operators are influenced by some kind of crowding in a way that no prior knowledge of the problem is required and no constraint affects either selection or replacement.

According to Sareni [8], among all the niching methods, fitness sharing (FS) is perhaps the most popular one. Introduced in [18], the idea is that an individual has only limited resources to share with others in the same niche. In spite of its proven worth, this technique has several drawbacks, e.g. it depends on the values of two parameters (the niche radius and the scaling factor), which cannot be easily determined. As a consequence, more advanced fitness sharing methods have been created such as implicit fitness sharing [27], where the fitness of each agent depends on the fitness values of the remaining individuals in the population, as well as the dynamic niche sharing (DNS) [7] and the dynamic fitness sharing (DFS) [2], both aiming at the self-identification of the niches in the population. Co-evolutionary sharing [14] tries to avoid the estimation of the niche radius while the sharing scheme based on niche identification techniques proposed in [10] is capable of locating the center and estimating the radius of each niche based on fitness topographical information.

Restricted tournament selection [17], clearing [4] and clustering [19] are other classical niching methods. The use of fuzzy logic in clearing and clustering respectively gave rise to the fuzzy clearing [34] and the genetic algorithm with sharing and fuzzy clustering (GASH-FC) [3]. In the latter approach, the number of niches is self-determined by constantly updating each niche radius until an optimal solution is found. Additional illustrative examples are sequential niching [11] and the species-conserving genetic algorithm (SCGA) by Li et al. [20].

Niching methods have also influenced Particle Swarm Optimization (PSO) –including NichePSO [26] and SPSO [16]–, as well as Differential Evolution (DE) –e.g. the species-based DE [35] and the Differential Evolution with an ensemble of Restricted Tournament Selection [9]. Besides, niching principles have been applied to combinatorial optimization problems [37, 28] and multi-objective ones too [32, 38].

Consistent efforts have been made in order to devise niching strategies in absence of parameters derived from *a priori* knowledge about the fitness landscape, like the *lbest* PSO niching algorithms using a ring topology [36] and the Adaptive Species Discovery [1] as well. Other techniques have been crafted to automatically approximate effective values for such parameters, e.g. the Adaptive Niching PSO [29] and the CMA-ES niching algorithm [25]. These schemes are concerned with what is perhaps the most common limitation encountered in the niching methods arena, i.e. their dependency on parameters which are often hard to set since prior knowledge about the problem is usually unavailable in real-world scenarios. Additionally, some niching techniques try to find solely all global optima, hence they ignore local ones. Finally, other drawbacks to be stated are that most niching algorithms perform defectively when the dimensionality of the problem or the number of optima increases, and some schemes cannot successfully preserve the previously found solutions.

Our proposed approach, NC-VMO, utilizes Pétrowski's classical niche clearing operator [4] as an addition to the original VMO formulation. This is consistent with the goal of this work, i.e. to corroborate VMO's capability to deal with multimodality instead of putting together a very competitive VMO multimodal scheme.

III. VARIABLE MESH OPTIMIZATION

This section briefly touches on the major algorithmic aspects of the VMO metaheuristic.

A. Algorithmic description

Variable Mesh Optimization is a population-based metaheuristic where solutions are represented in a mesh by a set of T nodes n_1, n_2, \dots, n_T . Each node is encoded as a vector of M real values $n_i(v_1^i, v_2^i, \dots, v_j^i \dots, v_M^i)$; $j = 1 \dots M$. The VMO algorithm is described by the pseudo code shown in Fig. 1, as a simplified version from that presented in [6]:

In this search process, the balance between exploration and exploitation is ensured by combining two phases: *expansion* and *contraction*. During the expansion, new nodes are generated toward the local and the global optima, and also from those in the mesh frontier. After that, the initial mesh for the next iteration is built in the contraction phase. VMO uses four main parameters: the initial population size (P) for all iterations, the maximum number of individuals (T) after the mesh expansion, the number of mesh nodes in the neighborhood (k) and the stop criterion (C).

B. Adaptive clearing operator

The VMO's adaptive clearing operator has a direct influence on its performance in multimodal problems as it is

discussed in Section VIII. In VMO, the contraction phase is in charge of selecting which individuals survive in the current iteration. Thus, based on an elitist strategy, nodes are sorted by their fitness values in such a way that survivor selection begins for that node with the best fitness. Before selection is over, an adaptive clearing operator is applied to the sorted nodes; they are sequentially compared and the worst among those closer (in fitness) than a threshold are deleted from the mesh. The value of such threshold for each of the M dimensions of the optimization problem is calculated as:

$$\varepsilon_j = \begin{cases} \text{range}(a_j, b_j)/2 & \text{if } c < 0.15C \\ \text{range}(a_j, b_j)/4 & \text{if } 0.15C \leq c < 0.3C \\ \text{range}(a_j, b_j)/8 & \text{if } 0.3C \leq c < 0.6C \\ \text{range}(a_j, b_j)/16 & \text{if } 0.6C \leq c < 0.8C \\ \text{range}(a_j, b_j)/100 & \text{if } c \geq 0.8C \end{cases} \quad (1)$$

where $\text{range}(a_j, b_j)$ are the domain boundaries j -th dimension ($j = 1 \dots M$), while c is the current control value related to the stop criterion C (e.g. number of iterations)

```

1. begin
2. Randomly generate  $P$  nodes for the initial mesh ( $3P \leq T$ )
3. Select the global best in the initial mesh
4. repeat
5.   for each node in initial mesh do
6.     Find its closest  $k$  nodes by their spatial locations
7.     Select the best neighbor as per the fitness values
8.     if current node is not the local best then
9.       Generate a new node toward the local best
10.    end if
11.  end for
12.  for each node in initial mesh but the global best do
13.    Generate a new node toward the global best
14.  end for
15.  Generate nodes from nodes in the mesh frontier (up to  $T$  nodes in the total mesh)
16.  Sort nodes according to their fitness values
17.  Apply the adaptive clearing operator
18.  Select  $P$  best nodes to build the initial mesh for the next iteration
19.  If needed, randomly generate new nodes so as to complete the initial mesh for the next iteration
20. until stop criterion ( $C$ ) is met
21. end

```

Figure 1. VMO's pseudo code.

IV. CLEARING-BASED NICHE FORMATION TECHNIQUE

Clearing [4] is a well-known procedure among niche formation methods, whose common denominator is the creation of stable subpopulations (niches) at all local optima (peaks) in the search space.

Every niche has a dominant (master) individual, i.e. the one with the best fitness. An individual belongs to a certain niche if its distance to the master is less than a given threshold known as the *clearing radius*. The method shares the *resources* of a niche among a fixed number of winners (individuals to be benefited by clearing), while it sets to zero the fitness of all other individuals in the same niche.

Depending on the fixed number of winners, the method might deal with more than a single one. In any case, the winner is the individual with the best fitness. Those subdued by the winner and the winner itself are fictitiously removed from the population. After repeating this procedure for a certain number of iterations, all winners emerge.

```

1. begin
2. Sort the population  $Pp$  in decreasing order of their fitness values
3. for  $i = 0$  to  $S - 1$ 
4.   if ( $\text{Fitness}(Pp[i]) \neq 0$ )
5.      $nbWinners = 1$ 
6.     Create a new niche, being  $Pp[i]$  its master
7.     for  $j = i + 1$  to  $S - 1$ 
8.       if ( $\text{Fitness}(Pp[j]) \neq 0$  and  $\text{Distance}(Pp[i], Pp[j]) < \sigma$ )
9.         if ( $nbWinners < \kappa$ )
10.           $nbWinners = nbWinners + 1$ 
11.          Insert individual  $Pp[j]$  in the current niche
12.        else
13.           $\text{Fitness}(Pp[j]) = 0$ 
14.        end if
15.      end if
16.    end for
17.  end if
18. end for
19. end

```

Figure 2. Pseudo code for the clearing-based niche formation technique.

The pseudo code for this niche formation algorithm is depicted in Fig. 2. Population Pp consists of an array of S individuals, σ denotes the clearing radius, κ is the niche capacity (maximum number of individuals) and $nbWinners$ is the number of winners of the subpopulation in the current niche.

V. NICHE-CLEARING-BASED VMO

Niche-Clearing-based Variable Mesh Optimization (NC-VMO) stems from applying the clearing procedure to VMO. Fig. 3 describes NC-VMO via pseudo code. Steps corresponding to the changes introduced are highlighted in *italic*. We discuss these modifications in Section VIII.

In NC-VMO, the contraction process starts (in step 16) by applying the niche formation technique (as detailed in Section IV), which operates over the previously expanded mesh. The identified niches are then affected (in step 19) by the VMO's adaptive clearing operator, described in Section III.B. Consequently, there is an evident difference between VMO and NC-VMO regarding when the adaptive clearing

operator is executed. In VMO, it involves the whole mesh at a time (step 17 in Fig. 1), but as shown in Fig. 3, this operator separately affects the niches discovered in the search process by NC-VMO. The rationale behind this algorithmic choice is given in Section VIII once the need clearly emerged after studying VMO's performance on different multimodal problems.

```

1. begin
2. Randomly generate  $P$  nodes for the initial mesh ( $3P \leq T$ )
3. Select the global best in the initial mesh
4. repeat
5.   for each node in the initial mesh do
6.     Find its closest  $k$  nodes by their spatial locations
7.     Select the best neighbor as per the fitness values
8.     if current node is not the local best then
9.       Generate a new node toward the local best
10.    end if
11.  end for
12.  for each node in initial mesh, but the global best do
13.    Generate a new node toward the global best
14.  end for
15.  Generate nodes from those in the mesh frontier (up to  $T$  nodes in the total mesh)
16.  Apply the clearing-based niche formation technique
17.  for each identified niche do
18.    Sort nodes according to their fitness values
19.    Apply the adaptive clearing operator
20.  end for
21.  Select  $P$  best nodes to build the initial mesh for the next iteration
22.  If needed, randomly generate new nodes so as to complete the initial mesh for the next iteration
23. until stop criterion ( $C$ ) is met
24. end

```

Figure 3. NC-VMO's pseudo code.

VI. ALGORITHMIC COMPLEXITY

In this section, a time complexity analysis for both VMO and NC-VMO is provided.

A. VMO's time complexity

Let n be the number of individuals in the population (in VMO, the P nodes in the initial mesh). The computational time required by VMO is shown in (2), where k and M are respectively the number of mesh nodes in the neighborhood and the dimension (number of variables) of the problem. Besides, t denotes the number of iterations of the algorithm.

$$time_{VMO}(n) = t[kn^2 + M[3n/2 + (3n - 1) + (3n - 2) + \dots + 1]] + 2n \quad (2)$$

Consequently, the time complexity for VMO is $O(tn^2)$, since t is a dominant coefficient, but neither k nor M .

B. NC-VMO's time complexity

Equation (3) describes the approximate time required by NC-VMO. Notice the introduction of q , which is the number of peaks maintained during the search. Also, qn [4] represents the cost of applying the clearing-based niche formation technique.

$$time_{NC-VMO}(n) = t[kn^2 + qn + qM[3n/2 + (n/q - 1) + (n/q - 2) + \dots + 1]] + 2n \quad (3)$$

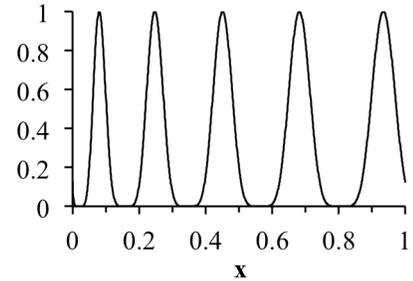
Despite the time required increases, the resulting global cost of NC-VMO is still in the same order of $O(tn^2)$.

VII. EXPERIMENTAL FRAMEWORK

A. Test functions

Among the large set of benchmark multimodal problems, the ones known in the literature as $F3$ and $F4$ are used as test functions in this work for two practical reasons. First, because among those problems available to study the behavior of niching methods, these two maximization problems are often used to evaluate the method's niche preservation ability rather than just assessing its search capabilities. The other reason is the need for comparison with previously published results on the behavior of other niching approaches, including those that applied the clearing niching procedure itself.

a. $F3$



b. $F4$

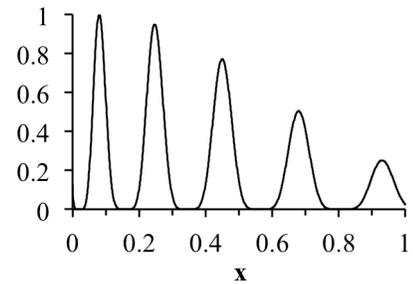


Figure 4. Test functions.

Equation (4) describes the analytical form of $F3$; it is graphically represented in Fig. 4a as well. This is a periodic function with peaks of equal size and interval.

$$F3(x) = \sin^6[5\pi(x^{0.75} - 0.05)] \quad (4)$$

Function $F3$ is defined for $x \in [0, 1]$ and its five peaks are located at $x = 0.080, 0.247, 0.451, 0.681$ and 0.934 , all with the same height of 1.

Equation (5) displays the analytical form of $F4$ while its graphical representation is shown in Fig. 4b. This is a periodic function having peaks of unequal size and width.

$$F4(x) = e^{-2(\log 2)\left(\frac{x-0.1}{0.8}\right)^2} \sin^6[5\pi(x^{0.75} - 0.05)] \quad (5)$$

Function $F4$ is defined for $x \in [0, 1]$ too, and it has also five peaks at $x = 0.080, 0.247, 0.451, 0.681$ and 0.934 , but they have different values of height, being approximately 1.000, 0.948, 0.770, 0.503 and 0.250 respectively.

B. Performance criteria

The following criteria have been used for assessing the performance of niching methods:

- *maximum peak ratio (MPR)*: indicates both the quality and the number of the optima identified. It is defined as the sum of the fitness of the optima reached by the niching technique divided by the sum of the fitness of the actual optima in the search space:

$$MPR = \frac{\sum_{i=1}^q F_i}{\sum_{j=1}^r f_j} \quad (6)$$

where F_i is the fitness of the found optimum i and f_j is the fitness of the actual optimum j . The variable q represents the number of identified peaks, which contain the discovered optima, while r is the number of real optima. An optimum is considered to be detected if its fitness value is at least 80% of the real one and the optimum is within a niche radius of the actual optimum. When an optimum is not detected, its fitness value is set to 0. Hence, the maximum value for the MPR is 1, when all optima are perfectly reached. [8, 16]

- *effective number of peaks maintained (ENPM)*: gives a measure of the ability of a niching method to locate and maintain individuals at the fitness peaks for extended periods of time. This is defined as a function of the number of evaluations after which stable values for the peaks found are reached. [8, 22]
- *chi-square-like deviation*: measures the deviation between the actual population distribution and an ideal proportionally populated distribution (μ_i, σ_i) in all the niches [7]. Thus, this performance indicator illustrates the ability of the method to proportionally populate the niches of the search space; the smaller the value, the better the method:

$$chi - square - like = \sqrt{\sum_{i=1}^{q+1} ((n_i - \mu_i)/\sigma_i^2)^2} \quad (7)$$

where, for the peak niches,

$$\mu_i = n f_i / \sum_{j=1}^q f_j \quad \text{and} \quad \sigma_i = \mu_i (1 - \mu_i / n) \quad (8)$$

and for the nonpeak niches:

$$\mu_{q+1} = 0 \quad \text{and} \quad \sigma_{q+1} = \sum_{i=1}^q \sigma_i^2 \quad (9)$$

n is the population size while f_i denotes the fitness value of the peak i . Besides, n_i denotes the experimental number of individuals in a niche i , μ_i corresponds to the expected ideal number and σ_i is the standard deviation of the number of individuals in the ideal distribution. [8]

C. Parametric configurations for VMO and NC-VMO

VMO's parameters are set as described in [6], except for the stop criterion. Thus, the initial population size (P) is set to 50 and the maximum number of nodes (T) after the expansion is 175, while for defining the neighborhoods $k = 3$ is used. To guarantee a fair comparison baseline with results reported by other methods, a maximum number of iterations ($C = 200$) is set as the stop condition. Each method is run 10 times for each test problem.

As an adapted VMO model, the new NC-VMO method uses the same four parameters P, T, k and C . Thus, in the case of NC-VMO they are also set as detailed above. In addition, for the clearing-based niche formation technique applied in NC-VMO, it is used $\kappa = 1$ as the niche capacity, since it is recommended by Pétrowski in [5] for the best performance. Besides, from results presented in [4], the number of winners (*nbWinners*) in a subpopulation is also set to 1. Hence, the masters receive the full resources of their corresponding niches. From the literature and due to topological reasons, $\sigma = 0.1$ is used as the niche radius for test problems $F3$ and $F4$.

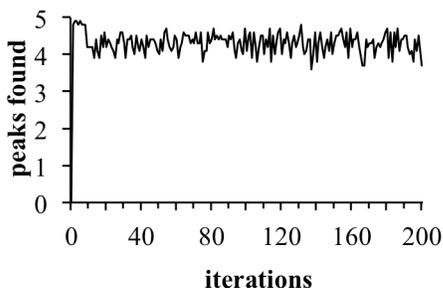
D. Baseline niching methods

In order to compare NC-VMO's performance to some outstanding GA's niching techniques, we refer to the results reported in [8] regarding the approximation of the same functions $F3$ and $F4$ by a few niching methods, including classical approaches. Such techniques are: the fitness sharing coupled with the matching sort algorithm (sharing+sort) [8], the clearing procedure, the deterministic crowding (DC) [31] and the restricted tournament selection (RTS) [17]. The results shown in [3] about the approximation of both test functions by the genetic algorithm with sharing and fuzzy clustering (GASH-FC) are also used for benchmarking.

VIII. VMO AND NC-VMO ON MULTIMODAL PROBLEMS

Average results over 10 independent runs of VMO's number of peaks found are reported in Fig. 5. Notice that VMO is capable of locating several of the desired peaks. It reached an average number of 4.3 peaks from those belonging to $F3$ and 4.5 in the case of $F4$. However, it is unable such a peak discovery rate for a long time. A preliminary analysis on the effect of the adaptive clearing operator reveals the reason of such inefficacy.

a. on $F3$



b. on $F4$

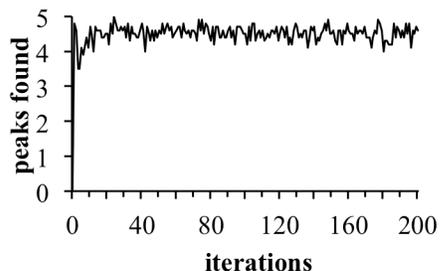


Figure 5. Mean number of peaks found by VMO.

As explained in Section III.B, as part of the selection process of the individuals surviving from the current iteration to the next, this operator eliminates the worst among those closer (in fitness) than a dynamic threshold. In order to illustrate the negative effect of such operator on the search of multiple optima (peaks), Table I shows the average number of nodes (in the *Nodes* column) located close to every peak at the end of the current iteration, as well as the average number of such nodes that get by the contraction process so as to be part of the initial mesh in the next iteration (in the *Survivors* column).

TABLE I. AVERAGE NUMBER OF NODES SURVIVING FROM THE CURRENT ITERATION TO THE NEXT

Peak	$F3$		$F4$	
	<i>Nodes</i>	<i>Survivors</i>	<i>Nodes</i>	<i>Survivors</i>
P_1	13.5	0	137.8	50
P_2	50.1	15	12	0
P_3	49.1	15	12.6	0
P_4	43.3	15	5.6	0
P_5	14.1	5	4.1	0

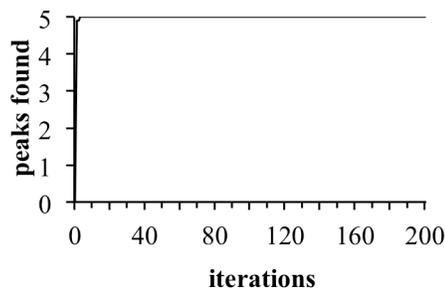
There is an obvious partial or total loss of nodes that approximate certain peaks. Thus, for some peaks the search is completely restarted whenever the population is renewed. This difficulty is caused by VMO's adaptive clearing operator. For instance, in the case of $F3$ none of the nodes in the surroundings of the first peak survive. This is because they have the worst fitness values among the 175 individuals in the population. Thus, sorted by their fitness, such nodes

will never be selected to be part of the mesh in the next iteration.

The consequences of the adaptive clearing operator are more drastic and evident on functions with peaks of unequal sizes, like $F4$. Note that all survivor nodes correspond to a single peak. The answer is in the height of such peak. As it represents the global optimum, the whole search process is focused on it, by approximating such solution by the average number of 137.8 nodes from the 175 in the total mesh. In addition, after sorting the nodes according to their fitness and removing the ones with inferior fitness among those closer than certain threshold, all the nodes that survive to the next iteration belong to the same best peak and the remaining ones have absolutely not a chance to continue existing.

To avoid this undesirable effect brought about by the adaptive clearing operator while preserving its power to foment diversity in the search, a new strategy is devised to deal with multimodality. As shown in Fig. 3, in the new algorithm NC-VMO, after the niche formation procedure is applied, the adaptive clearing operator independently comes into action in every niche found. It guarantees that at least the masters of all niches will be spared for the next iteration and therefore, all peaks are likely to be reached.

a. on $F3$



b. on $F4$

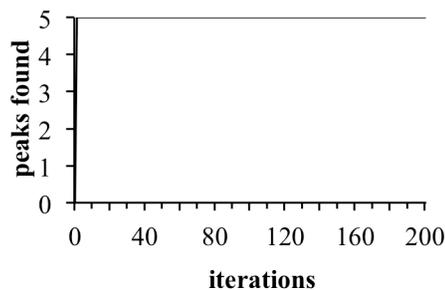


Figure 6. Mean number of peaks found by NC-VMO.

Consequently, as portrayed in Fig. 6, NC-VMO exhibits a great capacity to find and maintain all the desired peaks. It should also be noted its ability to rapidly converge to a maintained solution, as evidenced by the ENPM saturation values reached at the 3rd and 1st iterations for functions $F3$ and $F4$, respectively.

According to the fact that VMO is incapable of maintaining the reached peaks for extended periods of time,

it is not possible to talk about positive values for ENPM in this particular case. Then, as shown in Table II, the proposed method NC-VMO improves VMO regarding both the number of peaks maintained and the quality of the solutions. In addition, this new approach is also competitive with other niching methods from the literature. Based on the values reported for such techniques (see Table II), NC-VMO slightly outperforms both DC and sharing+sort. Also, together for $F3$ and $F4$, it behaves extremely close to the remaining GA's niching methods included in this study (clearing procedure, RTS and GASH-FC).

TABLE II. EFFECTIVE NUMBER OF PEAKS MAINTAINED (ENPM) AND MAXIMUM PEAKS RATIO (MPR)

Method	F3			F4		
	MPR	ENPM	chi-square	MPR	ENPM	chi-square
clearing	1	5	0.293	1	5	1.448
RTS	1	5	2.839	1	5	6.516
GASH-FC	1	5		1	5	
NC-VMO	0.999	5	0.024	0.999	5	0.060
sharing+sort	0.999	5	1.287	0.984	4.8	2.041
DC	1	5	5.683	0.778	4	9.582
VMO	0.837	0	0.043	0.896	0	0.106

The prior analysis is based on only the values of MPR and ENPM. However, a complementary evaluation on the values of chi-square-like deviation, also reported in Table II, exposes that VMO is superior to all the baseline methods concerning the capability for proportionally populate the desired niches. It might suggest the formation of stable subpopulations during the search. However, it has been proven that VMO is not able to create stable niches by losing some peaks from the current iteration to the next.

In addition, the values of the chi-square-like deviation corresponding to NC-VMO confirm its supremacy over VMO. The very low values of such performance statistic in this case illustrates that, also with results of MPR and ENPM very close to the best methods in comparison, NC-VMO is more stable than all the remaining methods regarding how capable they are on searching all the wanted optima in a parallel and proportional way. The most proportional the method populate the niches regarding the expected number of individuals in every niche, the most possible is to reach all the peaks.

Finally, it is interesting to dwell on the impact of reducing the population size for NC-VMO. As illustrated in Table III, the method still performs well when there are just 20 or even 10 nodes in the initial mesh. For a comparison with the baseline methods included in this study, it is worth mentioning that their population size is 100 individuals, except for GASH-FC that works with 150 chromosomes. It can be concluded that NC-VMO is still competitive with a smaller number of individuals; that is it has advantage with less population.

TABLE III. EXPERIMENTAL RESULTS OF NC-VMO FOR DIFFERENT POPULATION (MESH) SIZES

Mesh size		F3		F4	
P	T	MPR	ENPM	MPR	ENPM
50	175	0.999	5	0.999	5
30	105	0.999	5	0.999	5
20	70	0.998	5	0.999	5
16	56	0.998	5	0.998	5
10	35	0.994	5	0.995	5

IX. CONCLUSIONS AND FUTURE WORK

The performance evaluation of Variable Mesh Optimization on multimodal functions reveals that this metaheuristic holds promise for dealing with such kind of problems. VMO is capable of identifying a variety of local optima in the search space but it is not able to maintain them over time. As previously discussed, this is owing to the detrimental effect caused by its adaptive clearing operator, which is applied as part of its exploitation strategy.

To mitigate the difficulty described above, such strategy is modified by applying a clearing-based niche formation technique and the VMO's adaptive clearing operator, both working in a synergetic way. This is one of the first attempts to improve VMO's behavior in presence of multimodality. The resulting method, baptized as Niche-Clearing-based Variable Mesh Optimization, has proved its ability to find and retain multiple peaks as desired. Hybridizing VMO with a niching technique resulted indeed in an effective method to tackle multimodality.

As the applied clearing-based niche formation procedure relies on the niche radius, which requires *a priori* knowledge of the problem at hand, it is necessary to use other niching techniques that do not exhibit this drawback. Consequently, although the basic clearing niching procedure is successfully used in this study to demonstrate VMO's capacity to deal with multimodal problems, hybridization with other niching methods should improve the results achieved. Moreover, we ought to extend VMO's multimodal analysis to a broader set of problems that, for instance, have non-equidistant peaks. The niche radius could be, for example, dynamically updated as per the scheme proposed in [3].

ACKNOWLEDGEMENTS

The article has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 funded by Structural Funds of the European Union and state budget of the Czech Republic.

REFERENCES

- [1] A. Della Cioppa, A. Marcelli and P. Napoli, "Speciation in evolutionary algorithms: adaptive species discovery", Proc. 13th Genetic and Evolutionary Computation Conference (GECCO'11), Dublin, Ireland, 2011.
- [2] A. Della Cioppa, C. De Stefano and A. Marcelli, "Where are the niches? Dynamic fitness sharing" IEEE Transactions on Evolutionary Computation, vol. 11, no. 4, 2007, pp. 453-465.
- [3] A. El Imrani, A. Bouroumi, H. Zine El Abidine, M. Limouri and A. Essaid, "A fuzzy clustering-based niching approach to multimodal

- function optimization,” *Journal of Cognitive Systems Research*, vol. 1, 2000, pp. 119–133.
- [4] A. Pétrowski, “A clearing procedure as a niching method for genetic algorithms”, *Proc. 3rd IEEE International Conference on Evolutionary Computation*, Nagoya, Japan, 1996, pp. 798–803.
 - [5] A. Pétrowski, “A new selection operator dedicated to speciation”, *Proc. 7th Int’l Conference on Genetic Algorithms*, East Lansing, MI, 1997, pp. 144–151.
 - [6] A. Puris, R. Bello, D. Molina and F. Herrera, “Variable mesh optimization for continuous optimization problems”, *Soft Computing*, vol. 16, no. 3, 2011, pp. 511–525.
 - [7] B. L. Miller and M. J. Shaw, “Genetic algorithms with dynamic niche sharing for multimodal function optimization”, *Proc. IEEE Int’l Conference on Evolutionary Computation*, Piscataway, NJ, 1996, pp. 786–791.
 - [8] B. Sareni and L. Krähenbühl, “Fitness sharing and niching methods revisited”, *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 3, 1998, pp. 97–106.
 - [9] B.Y. Qu and P.N. Suganthan, “Novel multimodal problems and differential evolution with ensemble of restricted tournament selection”, *Proc. IEEE Congress on Evolutionary Computation*, Barcelona, Spain, 2010, pp. 3480–3486.
 - [10] C.Y. Lin and W.H. Wu, “Niche identification techniques in multimodal genetic search with sharing scheme”, *Advances in Engineering Software*, vol. 33, 2002, pp. 779–791.
 - [11] D. Beasley, D. R. Bull and R. R. Martin, “A sequential technique for multimodal function optimization”, *Evolutionary Computation*, vol. 1, no. 2, 1993, pp. 101–125.
 - [12] D. J. Cavicchio, “Adaptive search using simulated evolution”, Unpublished doctoral dissertation, University of Michigan, Ann Arbor, MI, 1970.
 - [13] D. Dilettoso and N. Salerno, “A self-adaptive niching genetic algorithm for multimodal optimization of electromagnetic devices”, *IEEE Transactions on Magnetics*, vol. 42, no. 4, 2006, pp. 1203–1206.
 - [14] D. E. Goldberg and J. Richardson, “Genetic algorithms with sharing for multimodal function optimization”, *Proc. 2nd Int’l Conference on Genetic Algorithms*, Cambridge, MA, 1987, pp. 41–49.
 - [15] D. E. Goldberg and L. Wang, “Adaptative niching via coevolutionary sharing”, in *Genetic Algorithms in Engineering and Computer Science*. New York: Wiley, 1997, pp. 21–38.
 - [16] D. Parrott and X. Li, “Locating and tracking multiple dynamic optima by a particle swarm model using speciation”, *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, 2006, pp. 440–458.
 - [17] G. R. Harik, “Finding multimodal solutions using restricted tournament selection”, *Proc. 6th International Conference on Genetic Algorithms*, Pittsburgh, PA, 1995, pp. 24–31.
 - [18] J.H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: The University of Michigan Press, 1975.
 - [19] J. Kennedy, “Stereotyping: Improving particle swarm performance with cluster analysis”, *Proc. IEEE International Conference on Evolutionary Computation*, 2000, pp. 303–308.
 - [20] J.P. Li, M.E. Balazs, G.T Parks and P.J. Clarkson, “A species conserving genetic algorithm for multimodal function optimization”, *Evolutionary Computation*, vol. 10, no. 3, 2002, pp. 207–234.
 - [21] K.A. De Jong, “An analysis of the behaviour of a class of genetic adaptive systems,” Doctoral dissertation, University of Michigan, Ann Arbor, MI, 1975.
 - [22] S. Das, S. Maity, B. Qu and P.N. Suganthan, “Real-parameter evolutionary multimodal optimization: survey of the state-of-the art”, *Swarm & Evolutionary Computation*, vol. 2, no. 3, 2011, pp. 71–88.
 - [23] O. Mengsheol and D. Goldberg, “Probabilistic crowding: Deterministic crowding with probabilistic replacement”, *Proc. 1st Genetic and Evolutionary Computation Conference (GECCO’99)*, 1999, pp. 409–416.
 - [24] O.M. Shir, C. Siedschlag, T. Bäck and M.J.J. Vrakking, “Niching in evolution strategies and its application to laser pulse shaping”, *Lectures Notes in Computer Science 3871*, Springer-Verlag Berlin, 2006, pp. 85–96.
 - [25] O.M. Shir and T. Bäck, “Niche radius adaptation in the cma-es niching algorithm”, *Proc. 9th Int’l Conference on Parallel Problems Solving from Nature*, Reykjavik, 2006, pp. 142–151.
 - [26] R. Brits and F. A. van den Bergh, “A niching particle swarm optimizer”, *Proc. 4th Asia-Pacific Conference on Simulated Evolution & Learning (SEAL’2002)*, Singapore, 2002, pp. 692–696.
 - [27] R. Smith, S. Forrest and A. S. Perelson, “Searching for diverse, cooperative populations with genetic algorithms”, *Evolutionary Computation*, vol. 1, no. 2, 1993, pp. 127–149.
 - [28] R. Yang, “Line-breeding Schemes for Combinatorial Optimization”, *Lectures Notes in Computer Science 1498*, Springer-Verlag Berlin, 1998, pp. 448–457.
 - [29] S. Bird and X.Li, “Adaptively choosing niching parameters in a PSO”, *Proc. 8th Genetic and Evolutionary Computation Conference (GECCO’06)*, Seattle, Washington, 2006, pp. 3–9.
 - [30] S. W. Mahfoud, “Crowding and preselection revisited”, *Proc. 2nd Int’l Conference on Parallel Problems Solving from Nature*, North-Holland, Amsterdam, 1992, pp. 27–36.
 - [31] S.W. Mahfoud, “Simple analytical models of genetic algorithms for multi-modal function optimization”, Technical report, Department of Computer Science, University of Illinois, Urbana, IL, Illinois Genetic Algorithm Laboratory Report No. 94005, 1993.
 - [32] T.Y. Chen and Y.S. Hsu, “A multiobjective optimization solver using rank-niche evolution strategy”, *Advances in Engineering Software*, vol. 37, 2006, pp. 68–699.
 - [33] W. Cedeño and V.R. Vemuri, “Analysis of speciation and niching in the multi-niche crowding GA”, *Theoretical Computer Science*, vol. 229, 1999, pp. 177–197.
 - [34] W. F. Sacco, M.D. Machado, C.M. Pereira and R. Schirru, “The fuzzy clearing approach for a niching genetic algorithm applied to a nuclear reactor core design optimization problem”, *Annals of Nuclear Energy*, vol. 31, 2004, pp. 55–69.
 - [35] X. Li, “Efficient differential evolution using speciation for multimodal function optimization”, *Proc. 7th Genetic and Evolutionary Computation Conference (GECCO’05)*, Washington, DC, 2005, pp. 873–880.
 - [36] X. Li, “Niching without niching parameters: particle swarm optimization using a ring topology”, *IEEE Transactions on Evolutionary Computation*, vol. 14, 2010, pp. 150–169.
 - [37] Y. Nagata, “Niching method for combinatorial optimization problems and application to JSP”, *Proc. IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, 2006.
 - [38] Z. Kowalczyk and T. Białaszewski, “Niching mechanisms in evolutionary computations”, *International Journal of Applied Mathematics and Computer Science*, vol. 16, no. 1, 2006, pp. 59–84.
 - [39] H. Izakian, A. Abraham, V. Snasel, *Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environments*, The 2009 IEEE International Workshop on HPC and Grid Applications (IWHGA2009), China, IEEE Press, USA, ISBN 978-0-7695-3605-7, pp. 8-12, 2009.
 - [40] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham and Bijaya Panigrahi, *Exploratory Power of the Harmony Search Algorithm: Analysis and Improvements for Global Numerical Optimization*, *IEEE Transactions on Systems Man and Cybernetics - Part B*, IEEE Press, USA, Volume 41, Issue 1, pp. 89-106, 2011.
 - [41] S. Das, A. Biswas, S. Dasgupta and A. Abraham, *Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications*, *Foundations of Computational Intelligence Volume 3: Global Optimization*, Studies in Computational Intelligence, Springer Verlag, Germany, ISBN: 978-3-642-01084-2, pp. 23-55, 2009.
 - [42] Hongbo Liu, Ajith Abraham and Maurice Clerc, *Chaotic Dynamic Characteristics in Swarm Intelligence*, *Applied Soft Computing Journal*, Elsevier Science, Volume 7, Issue 3, pp. 1019-1026, 2007.