Soft Comput
DOI 10.1007/s00500-011-0753-9

ORIGINAL PAPER

Variable mesh optimization for continuous optimization problems

Amilkar Puris · Rafael Bello · Daniel Molina · Francisco Herrera

Abstract Population-based meta-heuristics are algorithms that can obtain very good results for complex continuous optimization problems in a reduced amount of time. These search algorithms use a population of solutions to maintain an acceptable diversity level during the process, thus their correct distribution is crucial for the search. This paper introduces a new population meta-heuristic called "variable mesh optimization" (VMO), in which the set of nodes (potential solutions) are distributed as a mesh. This mesh is variable, because it evolves to maintain a controlled diversity (avoiding solutions too close to each other) and to guide it to the best solutions (by a mechanism of resampling from current nodes to its best neighbour). This proposal is compared with basic population-based meta-heuristics using a benchmark of multimodal continuous functions, showing that VMO is a competitive algorithm.

A. Puris · R. Bello
Department of Computer Science,
Universidad Central de las Villas, Santa Clara, Cuba
e-mail: ayudier@uclv.edu.cu

R. Bello
e-mail: rbellop@uclv.edu.cu

D. Molina (✉)
Department of Computer Languages and Systems,
University of Cadiz, Cadiz, Spain
e-mail: daniel.molina@uca.es

F. Herrera
Department of Computer Science and Artificial Intelligence,
University of Granada, Granada, Spain
e-mail: herrera@decsai.ugr.es

1 Introduction

Many real-world problems (advanced engineering design, data analysis, financial planning, risk management, scientific modelling, etc.) may be formulated as parameter optimization problems with variables in continuous domains, *continuous optimization problems*. Over the past few years, increasing interest has arisen in solving these kinds of problems (Lozano et al. 2011; Herrera and Lozano 2005; Michalewicz and Siarry 2008).

In recent years, a new family of search and optimization algorithms has been used to solve this kind of problem that does not offer a guarantee of locating an optimal solution, but gives satisfactory results for a much larger range of these problems. These algorithms extend basic heuristic methods by including them in an iterative framework to increase their exploration capabilities. This group of advanced approximate algorithms has been given the name of meta-heuristics and an overview of various existing methods is found in Glover and Kochenberger (2003). Meta-heuristics have proven to be highly useful for approximately solving difficult optimization problems in practice, because they may obtain good solutions in a reduced amount of time. Real-coded genetic algorithms (GAs) (Herrera et al. 1998), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), estimation of distribution algorithms (EDAs), scatter search (SS) (Laguna and Martí 2003) and difference evolution (DE) (Storn and Price 1997) are, among others, examples of the population-based meta-heuristics (PMHs). This term, PMHs, is used to group the meta-heuristics that use a solution set (called population) in each algorithm iteration.

PMHs introduce different ways of exploring the search space. They present powerful communication or cooperation mechanisms (depending on the context) to converge

Published online: 10 August 2011

the population toward promissory areas of the search space. This type of meta-heuristics implements several mechanisms to introduce diversity into the population and consequently makes a better exploration of different regions of the domain space possible, obtaining very good results in continuous search spaces (Lozano et al. 2011).

Another element that has a strong influence over the population search is the influence of the best solutions over the remainder. In real coded genetic algorithms (Herrera et al. 1998), the best individuals have a greater probability of survival and of having influence over the offspring; and, in other algorithms, the remaining solutions are oriented to the best ones directly, as in PSO (Kennedy and Eberhart 1995) or, more indirectly, as in DE (Storn and Price 1997; Brest et al. 2007).

With these two facts in mind, we come up with a new PMH called variable mesh optimization (VMO) for real parameter optimization. This model introduces a new mechanism to explore the search space, by creating more solutions around the most promising regions of the mesh. Another mechanism is used to foment population diversity by selecting the best representatives of the mesh for the next iteration. In this algorithm, the population is represented by a set of nodes (potential solutions) that are initially distributed as a mesh, using a uniform distribution. The search process developed by the VMO meta-heuristic can be described by two operations:

- Expansion: this mechanism explores around the best solutions found, by creating new nodes between each node of the mesh and its best neighbour, and around the external borders of the mesh.
- Contraction: this clearing process removes all nodes that are too close to others with best fitness. The aim is to maintain the population size and to foment mesh diversity.

We study, based on experimental work, the influence of its different components, showing that the ideas underlying this technique can lead to successful results. Then, we compare the performance of the VMO with other basic PMHs with multimodal functions on continuous domains, showing that VMO is a competitive model.

This paper is organized as follows: In Sect. 2, a detailed description of the VMO is given, emphasizing the expansion and contraction processes of the mesh. In Sect. 3, the experimental framework and the statistical tests used to validate the experimental results are presented. In Sect. 4, we analyse the behaviour of several VMO components. In Sect. 5, the proposed model is compared with others, to ascertain whether its innovative design is conducive to the outperformance of other PMHs. Finally, in Sect. 6, we present the main conclusions and suggest future work.

2 Variable mesh optimization

VMO is a PMH in which the population is distributed as a mesh. This mesh is composed of P nodes $(n_1, n_2, \ldots, n_P)$ that represent solutions in the search space. Each node is coded as a vector of M floating point numbers, $n_i = (v_1^i, v_2^i, \ldots, v_M^i) = v_j^i,\ j = 1, \ldots, M$ that represent the solution to the optimization problem. In the search process developed by VMO, two operations are executed: the expansion and contraction processes. Both processes introduce a suitable balance between exploration and diversity for the VMO algorithm. In following subsections, these operators are described in detail, and in Fig. 1 the

Fig. 1 Steps to apply the VMO algorithm

(Define input parameter setting: P, T, k and C, described in Table 1).
Step 1. P nodes are randomly generated to compose the initial population and all are evaluated to select the best n_g one.

Expansion

Step 2. For each node of the current population n_i (Z new nodes are created, $Z < P$):
2.1. find its closest k nodes using euclidian distance by Equation (1).
2.2. select the best node n_i^* (fitness) on neighbours.
2.3. if n_i^* is better than n_i, then
– calculate near factor Pr between n_i and n_i^* by Equation (3).
– generate a new node by Equation (2) using n_i, n_i^* and Pr.
Step 3. For each node n_i (X new nodes are created, $X = P - 1$):
3.1. calculate near factor Pr between n_i and n_g by Equation (3).
3.2. create a new node by Equation (6) using n_i, n_g and Pr.
Step 4. If $(Z + X) < T$, select the frontier nodes (external n_s and internal n_u nodes) (Y new nodes are created, $Y = T - (Z + X)$):
4.1. calculate w_j displacement vector by Equation (11).
4.2. generate $\lfloor Y/2 \rfloor$ new node from n_s by Equation (9).
4.3. generate $Y - \lfloor Y/2 \rfloor$ new node from n_u by Equation (10).

Contraction

Step 5. Merge and sort the current population with the created nodes in Steps 2-4 according to their fitness.
Step 6. Apply an adaptive clearing operator, eliminating the near nodes. We obtain B nodes for the next iteration.
Step 7. If $B < P$ then
– select the B nodes and complete the population with $P - B$ new random nodes.
Otherwise ($B >= P$) then
– select the P best nodes (fitness) to compose the population of the next iteration.

Step 8. Go to **Step 2** if the stop condition is not accomplished ($c < C$).

global algorithm is presented, using the parameters explained in Table 1.

2.1 Mesh expansion operation

The algorithm develops the expansion process by moving the population through the search space. For this action, new nodes are obtained, using the current population of each iteration, according to the following steps:

Step 1. (*Initial mesh generation*) The initial population for the first algorithm iteration is composed of P nodes that are randomly generated with uniform distribution.

Step 2. (*Nodes generation towards local extremes in the neighbourhood*) The first kind of exploration conducted in VMO is carried out in the neighbourhood of each node (n_i). The neighbourhood of n_i is composed of the k nodes closest (in terms of distance) to it. The best node (fitness) in the neighbourhood is selected as the local extreme (n_i^*). Only when n_i^* is better than n_i, a new node is generated between n_i and n_{i^*}. For this reason, in this step Z new nodes are created, where $Z \leq P - 1$.

To detect the neighbourhood of the i-th node, we used the euclidean distance (see Eq. 1)

$$D_{\text{euclidian}}(n_1, n_2) = \sqrt{\sum_{j=1}^{M} (v_j^1 - v_j^2)^2} \tag{1}$$

The new nodes (n_z) are calculated using the function F defined by Eq. 2

$$n_z = F(n_i, n_i^*, Pr(n_i, n_i^*)) \tag{2}$$

where Pr is the near factor and represents the relation between the fitness of the current node and its local extreme. This factor is calculated by Eq. 3. It takes a value in the range [0, 1], higher when better fitness has n_i.

$$Pr(n_i, n_i^*) = \frac{1}{1 + |\text{fitness}(n_i) - \text{fitness}(n_i^*)|} \tag{3}$$

The function F can be described in different ways. For this study, the components v_j^z of the new nodes n_z are calculated by Eq. 4:

$$v_j^z = \begin{cases} \overline{m_j}, & \textbf{if } |\overline{m_j} - v_j^{i^*}| > \xi_j \text{ and } U[0,1] \leq Pr(n_i, n_i^*) \\ v_j^{i^*} + U[-\xi_j, \xi_j], & \textbf{if } |\overline{m_j} - v_j^{i^*}| \leq \xi_j \\ U[v_j^i, \overline{m_j}], & \textbf{othercase} \end{cases} \tag{4}$$

where $\overline{m_j} = \text{average}(v_j^i, v_j^{i^*})$, $U[x, y]$ denotes a random value (uniformly) in the interval $[x, y]$, and ξ_j defines the minimum allowed distance for each component. Its value decreases during the running of the algorithm, calculated by Eq. 5,

$$\xi_j = \begin{cases} \frac{range(a_j,b_j)}{4} & \text{if } c < 0.15\%C \\ \frac{range(a_j,b_j)}{8} & \text{if } 0.15\%C \leq c < 0.3\%C \\ \frac{range(a_j,b_j)}{16} & \text{if } 0.3\%C \leq c < 0.6\%C \\ \frac{range(a_j,b_j)}{50} & \text{if } 0.6\%C \leq c < 0.8\%C \\ \frac{range(a_j,b_j)}{100} & \text{if } c \geq 0.8\%C \end{cases} \tag{5}$$

where C and c denote a maximum number of fitness evaluations allowed and the current number of fitness evaluations. In addition, the $range(a_j, b_j)$ denotes the domain amplitude (a_j, b_j) of each component.

F function behaves as follows: in the first case, the average value between the current node and the local extreme is obtained for the j-th component. In the second case, the local extreme neighbourhood is displaced depending on a distance value for the current iteration. In the last case, a random number is generated between the average value and the current node.

Figure 2 shows an example of this step, with $k = 4$.

Step 3. (*Nodes generation towards the global extreme*) This step is used to accelerate the algorithm convergence. For this, it explores in the direction of the node which has the best fitness of the current population, called the global extreme of the mesh (n_g). X new nodes are generated ($X = P - 1$), one for each n_i towards n_g using Eq. 6.

$$n_x = G(n_i, n_g, Pr(n_i, n_g)) \tag{6}$$

Table 1 Parameters of the VMO algorithm

Parameter	Description
P	Number of nodes of the population for each iteration
T	Number of new nodes required in the expansion process
k	Number of nodes that define the neighbourhoods of each node of the mesh
C	Algorithm stop condition (maximum number of fitness evaluations)

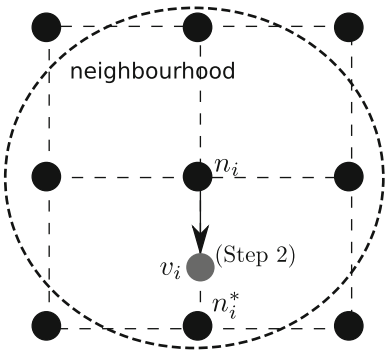

Fig. 2 Example of Step 2

In Eq. 7 we present the G function used in this work, where v_j^x represents the values computed for each component of the new nodes n_x.

$$v_j^x = \begin{cases} \text{average}(v_j^i, v_j^g), & \text{if} \quad U[0,1] \leq Pr(n_i, n_g) \\ U[\text{average}(v_j^i, v_j^g), v_j^g], & \text{otherwise} \end{cases} \tag{7}$$

If there is a great difference (in fitness) between n_i and n_g, then there will be a high probability for the j-th component to take closer values to v_j^g. In the other case, the average value is obtained.

Figure 3 shows a example of this step.

Step 4. (*Nodes generation starting from the frontier nodes of mesh*) In this step, Y new nodes are created from the frontier nodes in the current population to complete the expansion process. This step is only run if the number of created nodes in the previous steps ($Z + X$) is lower than T, and $Y = T - (Z + X)$ are created in this step. If $Y > P$, only P new nodes are created, one for each mesh node.

In this step, we consider frontier nodes (n_t). The frontier is composed of the nodes that are nearest (*known as interior frontier or internal nodes*, n_u) and furthest (*known as exterior frontier or external nodes*, n_s) from the point that represents the search space center. To detect these nodes, the euclidean distance is used. The nodes of the highest distance compose the set n_s and the ones with the smallest distance compose the set n_u. Starting from these sets, new nodes are created (one for each frontier node) using the function H (see Eq. 8).

$$n_h = H(n_t, w) \tag{8}$$

For our study, the function H is defined by means of the Eqs. 9 and 10, where the components v_j^h of each new node n_h are calculated depending on the frontier type:

In Eq. (9) $\lfloor Y/2 \rfloor$ nodes are created, using n_s set:

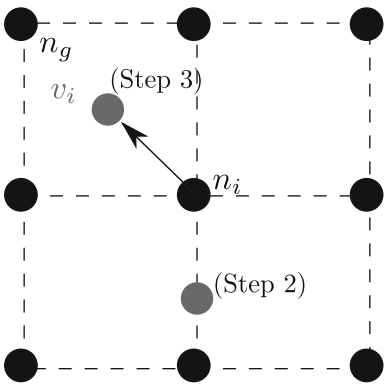

Fig. 3 Example of Step 3

$$v_j^h = \begin{cases} v_j^s + w_j, & \text{if} \quad v_j^s > 0 \\ v_j^s - w_j, & \text{if} \quad v_j^s < 0 \end{cases} \tag{9}$$

In Eq. (9) $Y - \lfloor Y/2 \rfloor$ nodes are created, using n_u set:

$$v_j^h = \begin{cases} |v_j^u + w_j|, & \text{if} \quad v_j^u > 0 \\ |v_j^u - w_j|, & \text{if} \quad v_j^u \leq 0 \end{cases} \tag{10}$$

where w_j represents a displacement for each component j and is calculated in a decreasing way according to Eq. 11:

$$w_j = (w_j^0 - w_j^1) \cdot \frac{C - c}{C} + w_j^1 \tag{11}$$

The variable w_j^0 represents the initial displacement and w_j^1 its final value. In addition, the values C and c are used (see the description of Eq. 5). In order to obtain decreasing displacements $w_j^0 > w_j^1$ and its values are calculated as: $w_j^0 = range(a_j, b_j)/10$ and $w_j^1 = range(a_j, b_j)/100$.

Figure 4 shows an example of this step, with $Y = 4$.

2.2 Mesh contraction process

The contraction operation selects the nodes of the mesh that will be used as the population for the next algorithm iteration. Nodes with the best fitness are selected from among the current mesh and the new nodes created for the expansion process. In this way, it is similar to the elitist selection operators in evolutionary algorithms (Deb 2001; Herrera et al. 1998). Before the selection, a method to increase diversity in the mesh is presented to keep a minimum distance between the mesh nodes. To achieve this, an adaptive clearing operator is proposed. In the following, the contraction process is described:

Step 5. All mesh nodes are ordered depending on their fitness (ascendant).

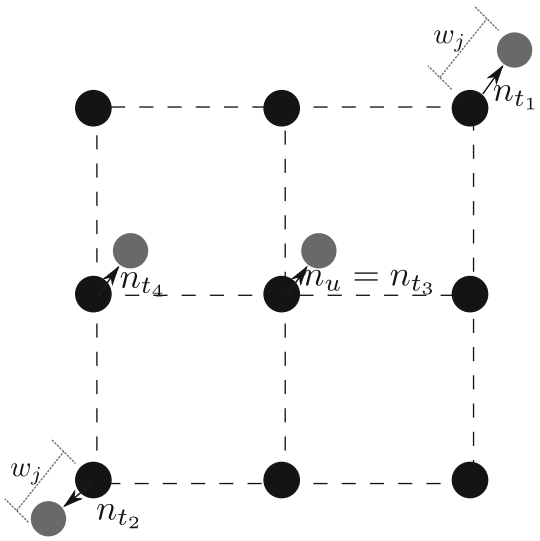

Fig. 4 Example of Step 4

Step 6. The difference between each node and their successors is calculated for each dimension. Successor nodes with any difference smaller than its corresponding ξ_j value are removed. The ξ_j value is calculated by Eq. 5. This strategy is called adaptive clearing, and finally we are left with B nodes.

Step 7. The nodes with best fitness are selected as the population for the next iteration. If $B < P$ then the mesh is completed with new random nodes.

This mechanism considers two important elements: the node qualities and their places in the solution space. The nodes with better fitness have a higher probability of taking part in the next population. Adaptive clearing allows the method to carry out more general explorations and eventually to reduce its frequency to focus on a smaller search space area. This element increases the method's exploitation level and makes it stronger.

3 Experimental framework

We have carried out different experiments to assess the performance of VMO. In this section, we describe the test functions (Sect. 3.1), the statistical methods (Sect. 3.2) that were used to validate the results, and the experimental setup (Sect. 3.3).

3.1 Test functions

The test suite that we have used for the experiments consists of 20 benchmark multimodal functions chosen from the set designed for the special session on real parameter optimization organized in the 2005 IEEE congress on evolutionary computation (CEC2005) (Suganthan et al. 2005). The unimodal functions are not used for this study because these are simpler than the multimodal functions. In Suganthan et al. (2005), the complete description of the multimodal functions and their source codes can be found. The set of test functions is composed of the following functions:

- seven basic multimodal functions (F_6–F_{12}):
  - Shifted Rosenbrock's function.
  - Griewank's function, displaced and rotated without frontiers.
  - Ackley's function, displaced and rotated with the global optimum in the frontier.
  - Rastrigin's function, displaced.
  - Rastrigin's function, displaced and rotated.
  - Weierstrass' function, displaced and rotated.
  - Schwefel's problem 2.13.
- Two expanded multimodal functions (F_{13} and F_{14}).
- Eleven hybrid functions (F_{15}–F_{25}). Each one of them has been defined through combination of 10 out of the 14 previous functions (different in each case).

All functions were displaced to ensure that their optima can never be found in the center of the search space. In two functions, in addition, the optima cannot be found within the initialization range, and the domain of the search is not limited (the optimum is out of the initialization range).

3.2 Statistical methodology for comparisons

When a new optimization algorithm proposal is developed, it is necessary to compare it with previous approaches. Statistical analysis needs to be carried out to find significant differences among the results obtained by the studied methods. In García et al. (2009a, b), the uses of some nonparametric tests (Sheskin 2007) are presented to compare the results in computational intelligence. In particular, we have considered two alternative methods based on nonparametric tests to analyse the experimental results:

- Application of Iman and Davenport's test (1980) and Holm's method (1979) as post hoc procedures. The first test may be used to see whether there are significant statistical differences among the algorithms in certain groups (three or more algorithms). If differences are detected, then Holm's test is employed to compare the best ranking algorithm (control algorithm) with the remaining ones.
- Utilization of the Wilcoxon (1945) matched-pairs signed-ranks test. With this test, the results of two algorithms may be directly compared.

Any reader interested in this topic can find additional information on the Web site http://sci2s.ugr.es/sicidm.

3.3 The experimental setup

The experiments have been developed in the two following ways:

- First, we analyse the influence of any parameters and their components on the VMO.
  - Size of mesh: VMO is tried with different sizes of the initial population to test if this has a strong influence on the quality of results.
  - Adaptive clearing operator: the influence of the adaptive clearing on the behaviour of the VMO is tested.
  - The expansion towards the frontiers (h function): the influence on the results of the exploration operator across the frontier nodes is observed.

– Then, we compare the VMO results with other algorithms from the literature.

All experiments have been carried out with dimension $D = 10$, $D = 30$ and $D = 50$, and the maximum number of fitness evaluations (stopping criterion) that we allowed for each algorithm is $C = 10,000 \cdot D$. The value of the k parameter used was $k = 3$ for all experiments and the other parameters were defined in each test suite. Each algorithm is run 25 times for each test function, and the average error of the best found solution is computed. The function error value for a solution x is defined as $(f(x_j) - f(x_j^*))$, where x_j^* is the global optimum of the function. The results obtained in each experimental case are executed using the same test suite and are shown in Appendix as tables.

4 Internal analysis of VMO

In this section, we study the VMO's behaviour for some internal elements: size of the initial population (Sect. 4.1), adaptive clearing operator (Sect. 4.2) and generation by means of the frontier's nodes (Sect. 4.3).

4.1 Size of mesh

The definition of the size of the population is very important in current PMHs to obtain a high performance from these methods. For instance, in the GAs the populations must be big enough (Minetti 2005); due to the fact that not all the individuals influence the evolution of the population, a process of selection is conducted in each iteration to obtain a subgroup of the population to generate new individuals. In the specific case of the PSO (Kennedy and Eberhart 1995), use of smaller populations is recommended (Engelbrecht 2006) because all the agents change their position, directly influencing their displacement in the flock.

The process presented by the VMO is similar to PSO, in which all the population nodes are involved in the exploration. Due to this similarity, different sizes of mesh are studied: $P = (8, 12, 24, 50$ and $100)$. In all cases, the total expansion size used for these studies is $T = \frac{3}{2}P$. The results for the test functions are shown in Appendix Tables 14, 15 and 16.

To apply the nonparametric test to this case study, we present in Table 2 the average ranking of the VMO algorithm for different mesh sizes for all dimensions (the VMO(P) refers to the VMO algorithm with P population size, the best rank is presented in bold typeface). The results of Iman–Davenport's test show that the mesh size could produce significant differences; due to this fact, the hypothesis of equality has been rejected for each dimension (see Table 3). This conclusion was obtained because the statistical test value was greater than the critical value, 2.4920 in this case.

Table 2 Ranking of the VMO algorithm for different mesh sizes for each dimension

Algorithm	Rank ($D = 10$)	Rank ($D = 30$)	Rank ($D = 50$)
VMO(8)	4.4750	3.449	2.875
VMO(12)	3.650	**1.400**	**1.400**
VMO(24)	2.900	2.600	2.775
VMO(50)	**1.651**	3.950	3.250
VMO(100)	2.325	3.600	4.700

Table 3 Results of Iman–Davenport's test for different mesh sizes for each dimension

Dimension	Test value	Critical value	Hypothesis
$D = 10$	18.154	2.4920	Rejected
$D = 30$	13.674	2.4920	Rejected
$D = 50$	23.974	2.4920	Rejected

To continue, Holm's test is applied to compare the configuration with the best rank in each dimension (VMO(50) for $D = 10$ and VMO(12) for $D = 30$ and $D = 50$), with each of the four remaining ones. For this test, the algorithms are ordered in descending order according to rank.

Table 4 contains all the computations associated with Holm's procedure ($z = (R_0 - R_i)/SE$, p value, and α/i) for a standard error for each dimension of $SE = 0.5$. The last column indicates whether the control algorithm performs significantly better than the corresponding algorithm (first column), in this case the equality hypothesis is rejected. This conclusion is arrived at because the p values are smaller than the corresponding α/i values. For this comparison, the results obtained with a mesh size of 50 nodes are significantly superior to three of the compared configurations (VMO(8), VMO(12) and VMO(24)) for $D = 10$. Only in comparison with VMO(100) are the differences insignificant. For $D = 30$ and $D = 50$, the VMO algorithm with a population size of 12 nodes obtained significantly superior results to the other mesh configurations.

We have applied Wilcoxon's test to compare VMO(50) configuration with VMO(100) for $D = 10$. In this test, the values of R^- and R^+ (associated with the control algorithm in comparison) are specified (the lowest ones, which correspond to the worst results, are highlighted in bold face), together with the p values computed for this test and whether the hypothesis is rejected (the p value is lower than the significance value) or not. Table 5 shows the results, showing that that VMO(50) obtained better results than

Table 4 Results of the Holm's test for each dimension with a significance value 0.05

i	Algorithm	$z = (R_0 - R_i)/SE$	p value	α/i	Hypothesis
$D = 10$, VMO(50) as reference algorithm					
4	VMO(8)	5.649	1.60E−08	0.0125	Rejected
3	VMO(12)	3.999	6.33E−05	0.0166	Rejected
2	VMO(24)	2.499	0.012	0.0250	Rejected
1	VMO(100)	1.349	0.177	0.0500	Accepted
$D = 30$, VMO(12) as reference algorithm					
4	VMO(50)	5.100	3.39E−7	0.0125	Rejected
3	VMO(100)	4.400	1.08E−5	0.0166	Rejected
2	VMO(8)	4.099	4.13E−5	0.0250	Rejected
1	VMO(24)	2.400	0.016	0.0500	Rejected
$D = 50$, VMO(12) as reference algorithm					
4	VMO(100)	6.600	4.11E−11	0.0125	Rejected
3	VMO(50)	3.699	2.15E−4	0.01666	Rejected
2	VMO(8)	2.949	0.003	0.0250	Rejected
1	VMO(24)	2.750	0.005	0.0500	Rejected

Table 5 Results of Wilcoxon's test (significance value, 0.05)

VMO(50) vs	R^+	R^-	p value	Hypothesis
VMO(100)	138.00	**72.00**	0.157	Accepted

VMO(100) (the R^+ values are higher than the R^- ones). But this difference is not statistically significant (because $0.157 > 0.05$). Because VMO(50) is the best mesh size, it will used in the studies for dimension 10.

In these experiments, we can conclude that the mesh size has a relation to the dimension of the problem. For problems with a small dimension, a population size of between 50 and 100 nodes obtains good results, and for high dimensions the best results are obtained with a mesh of 12 nodes.

4.2 Adaptive clearing operator

Here, we study the effect of applying an adaptive clearing operator to the VMO algorithm. For this study, we compare the results among other clearing operators and our adaptive proposal. The results of this comparison are shown in Appendix Tables 17, 18 and 19. VMO-NC and VMO-AC represent the solutions to the algorithm, without the clearing operator and using the adaptive clearing, respectively. VMO-C1, VMO-C2, VMO-C3, VMO-C4 and VMO-C5 show the VMO solutions with the clearing operator for different constant values of the distance ($\xi_j = \frac{range(a_j,b_j)}{4}, \frac{range(a_j,b_j)}{8}, \frac{range(a_j,b_j)}{16}, \frac{range(a_j,b_j)}{50}$ and $\frac{range(a_j,b_j)}{100}$, respectively), depending of the domain amplitude in each test function (see Eq. 6).

In Table 6, we show the mean rankings of each executed alternative for different clearing operators.

Table 6 Ranking of VMO's algorithm with different clearing operators for each dimension

Algorithm	Rank ($D = 10$)	Rank ($D = 30$)	Rank ($D = 50$)
VMO-NC	6.449	6.649	5.850
VMO-C5	5.525	5.625	4.575
VMO-C4	4.575	4.975	4.700
VMO-C1	4.200	3.900	4.024
VMO-C3	2.825	3.125	4.300
VMO-C2	2.575	2.600	3.250
VMO-AC	1.850	1.125	1.300

Table 7 Results of the Iman–Davenport's test for different alternatives of clearing operator for each dimension

Dimension	Test value	Critical value	Hypothesis
$D = 10$	26.315	2.1791	Rejected
$D = 30$	64.320	2.1791	Rejected
$D = 50$	14.641	2.1791	Rejected

First, we apply Iman–Davenport's test to each dimension. Table 7 shows the results, detecting statistically significant differences for each dimension. Thus, we applied Holm's multiple comparisons test to find out which algorithm was statistically better than the others.

Holm's test compares the algorithm with the best rank for each dimension VMO-AC, with each one of the other configurations, in pairs. Table 8 shows the results of Holm's test with the significance value 0.05. We can see that there are significant differences in the majority of cases, with the exception of VMO-C3 and VMO-C2, where there are no significant differences in dimension 10.

Table 8 Results of the Holm's test for each dimension

i	Algorithm	$z = (R_0 - R_i)/SE$	p value	α/i	Hypothesis
$D = 10$, VMO-AC as reference algorithm					
6	VMO-NC	6.733	1.65E−11	0.0080	Rejected
5	VMO-C5	5.379	7.46E−8	0.0100	Rejected
4	VMO-C4	3.989	6.63E−5	0.0125	Rejected
3	VMO-C1	3.440	5.86E−4	0.0166	Rejected
2	VMO-C3	1.427	0.153	0.0250	Accepted
1	VMO-C2	1.061	0.288	0.0500	Accepted
$D = 30$, VMO-AC as reference algorithm					
6	VMO-NC	8.087	6.07E−16	0.0080	Rejected
5	VMO-C5	6.587	4.48E−11	0.0100	Rejected
4	VMO-C4	5.635	1.74E−8	0.0125	Rejected
3	VMO-C1	4.062	4.86E−5	0.0170	Rejected
2	VMO-C3	2.927	0.003	0.0250	Rejected
1	VMO-C2	2.159	0.031	0.0500	Rejected
$D = 50$, VMO-AC as reference algorithm					
6	VMO-NC	6.661	2.73E−11	0.0080	Rejected
5	VMO-C4	4.977	6.45E−7	0.0100	Rejected
4	VMO-C5	4.794	1.63E−6	0.0125	Rejected
3	VMO-C3	4.391	1.13E−5	0.0167	Rejected
2	VMO-C1	3.989	6.65E−5	0.0250	Rejected
1	VMO-C2	2.855	0.004	0.0500	Rejected

Table 9 Results of Wilcoxon's test (significance value 0.05)

VMO-AC vs	R^+	R^-	p value	Hypothesis
VMO-C3	156.50	**53.50**	0.038	Rejected
VMO-C2	167.50	**42.50**	0.013	Rejected

We have also applied Wilcoxon's test to determine which of them presents the best behaviour. Table 9 shows the results between VMO-AC and VMO-C3, and VMO-AC and VMO-C2 for $D = 10$. In both cases, the differences in favour of VMO-AC (R^+ values are higher than R^-) are statistically significant. VMO-AC is the best algorithm, proving that a clearing method improves the results, and the proposed adaptive clearing method statistically improves results obtained with a fixed distance value. Thus, in the following, the VMO uses the adaptive clearing method.

4.3 Generation from the frontier nodes

This generation process explores the domain space around the frontiers of the population. For this reason, it is necessary to carry out an experimentation to determine whether the process should be applied. Then, we will present a statistical study of the experimental results shown in Table 20 in the Appendix, in which the column VMO-NF represents the algorithm VMO when the frontier's operator is not used, and VMO-F when it is used. For this study, we use Wilcoxon's test to compare both algorithms for each dimension.

Table 10 Results of Wilcoxon's test (significance value 0.05)

VMO-F vs	R^+	R^-	p value	Hypothesis
$D = 10$				
VMO-NF	190.50	**19.50**	0.001	Rejected
$D = 30$				
VMO-NF	180.50	**29.50**	0.002	Rejected
$D = 50$				
VMO-NF	180.50	**29.50**	0.002	Rejected

It can clearly be seen (see Table 10) that VMO-F obtains better results than VMO-NF in all dimensions (R^+ values are higher than the R^- ones). In addition, the statistical test indicates that these improvements are statistically significant.

In the following, the algorithm VMO will be used with the adaptive clearing operator and the nodes generation operator starting from the frontiers.

5 Comparative study with others algorithms

A comparative study is conducted between VMO and other PMHs that have a demonstrably high level of performance. These algorithms belong to evolutionary algorithms and

swarm Intelligence. The local search algorithms are not used to improve the solutions. To continue, a brief description of the methods is presented:

– Steady-state genetic algorithm (SSGA) (Syswerda 1989), with an election of parents negative assortative mating (Fernandes and Rosa 2001) with $N_{nam} = 3.$, and a replacement strategy of replacing the worst.
– Linearly decreasing inertia weight in particle swarm optimization (LDWPSO) (Shi and Eberhart 1998): the algorithm PSO proposed by Shi and Ebenhart is taken into account, applying the authors' configuration: the inertia varies from the maximum value ($w_{\max} = 0.9$) to the minimum value ($w_{\min} = 0.4$); the parameters c1 and c2 are equal to 2.8 and 1.3, respectively.
– Opposite differential evolution (ODE) (Rahnamayan et al. 2008): this algorithm is a DE that enforces diversity in the search, considering in the search process the opposite of each new solution created.

The parameters used in all cases were defined by the authors themselves.

We present a statistical analysis conducted over the results shown in Appendix Tables 23, 24 and 25. In this analysis, we compare VMO with other algorithms presented in this section, using Wilcoxon's test.

According to Wilcoxon's test results summarized in Tables 11, 12 and 13 it can be observed that:

– VMO is significantly better than the SSGA and LDWPSO for $D = 10$. The statistical test indicates that these improvements are statistically significant. VMO is only worse in absolute terms in relation to ODE ((R^+ values are lower than the R^- ones)), but not in a significant way.

Table 11 Results of Wilcoxon's test for $D = 10$

VMO vs	R^+	R^-	p value	Hypothesis
SSGA	210.0	**0.0**	0.000	Rejected
ODE	**89.0**	121.0	0.550	Accepted
LDWPSO	210.0	**0.0**	0.000	Rejected

Table 12 Results of Wilcoxon's test for $D = 30$

VMO vs	R^+	R^-	p value	Hypothesis
SSGA	210.0	**0.0**	0.000	Rejected
ODE	156.5	**53.5**	0.040	Rejected
LDWPSO	210.0	**0.0**	0.000	Rejected

Table 13 Results of Wilcoxon's test for $D = 50$

VMO vs	R^+	R^-	p value	Hypothesis
SSGA	210	**0**	0.000	Rejected
ODE	205	**5**	0.000	Rejected
LDWPSO	167	**43**	0.044	Rejected

Table 14 Results for different mesh sizes for dimension 10

Function	VMO(8)	VMO(12)	VMO(24)	VMO(50)	VMO(100)
F6	1.32E+02	4.22E+01	7.15E+01	6.02E+01	8.41E+01
F7	4.15E+03	3.79E+03	3.91E+03	6.39E+02	4.00E+03
F8	2.04E+01	2.04E+01	2.04E+01	2.03E+01	2.03E+02
F9	1.49E+01	1.28E+01	1.12E+01	3.10E+00	9.68E+00
F10	2.11E+01	1.51E+01	7.00E+00	4.79E+00	7.04E+00
F11	5.21E+00	2.74E+00	2.48E+00	5.12E+00	2.47E+00
F12	6.53E+02	1.04E+02	1.60E+02	7.55E+01	9.97E+01
F13	1.73E+00	1.57E+00	7.81E−01	7.48E−01	7.60E−01
F14	2.93E+00	2.75E+00	2.57E+00	2.74E+00	2.55E+00
F15	3.34E+02	2.59E+02	2.95E+02	2.14E+02	2.24E+02
F16	1.45E+02	1.27E+02	1.14E+02	1.03E+02	1.04E+02
F17	1.53E+02	1.42E+02	1.29E+02	1.20E+02	1.30E+02
F18	8.26E+02	8.51E+02	8.27E+02	7.46E+02	7.50E+02
F19	8.06E+02	8.62E+02	8.00E+02	6.98E+02	7.49E+02
F20	8.49E+02	8.55E+02	7.99E+02	7.87E+02	7.52E+02
F21	6.70E+02	7.48E+02	6.79E+02	5.53E+02	5.27E+02
F22	8.21E+02	7.91E+02	7.75E+02	7.51E+02	7.61E+02
F23	9.72E+02	8.78E+02	8.84E+02	6.45E+02	7.32E+02
F24	4.16E+02	4.12E+02	4.16E+02	4.06E+02	3.90E+02
F25	4.13E+02	3.98E+02	4.00E+02	3.88E+02	3.65E+02

– VMO is statistically better than all the alternatives taken into consideration for the $D = 30$. The difference in respect to ODE increases.
– Finally, for dimension 50, results obtained by VMO are significantly better than all the PMHs it was compared with.

Table 15 Results for different mesh sizes for dimension 30

Test function	VMO(8)	VMO(12)	VMO(24)	VMO(50)	VMO(100)
F6	9.12E+02	2.82E+02	1.98E+03	1.75E+03	2.15E+03
F7	8.13E+03	2.72E+02	7.94E+03	2.79E+04	2.98E+04
F8	2.10E+01	2.09E+01	2.09E+01	2.10E+01	2.10E+01
F9	1.23E+02	1.04E+02	9.69E+01	1.32E+02	1.12E+02
F10	5.15E+01	5.31E+00	5.08E+01	6.15E+01	5.72E+01
F11	1.50E+01	1.41E+01	1.77E+01	1.60E+01	1.56E+01
F12	1.72E+04	9.56E+03	1.34E+04	1.37E+04	1.25E+04
F13	4.87E+00	4.52E+00	4.37E+00	5.06E+00	4.57E+00
F14	1.21E+01	1.01E+01	1.17E+01	1.24E+01	1.19E+01
F15	3.59E+02	3.36E+02	3.30E+02	3.78E+02	3.54E+02
F16	2.10E+02	2.68E+02	1.58E+02	2.40E+02	2.03E+02
F17	2.86E+02	1.97E+02	2.36E+02	3.21E+02	2.18E+02
F18	9.30E+02	8.07E+02	8.98E+02	9.31E+02	9.37E+02
F19	8.30E+02	8.06E+02	9.40E+02	8.32E+02	9.37E+02
F20	8.60E+02	8.06E+02	8.38E+02	8.31E+02	8.36E+02
F21	7.05E+02	5.00E+02	7.20E+02	7.77E+02	7.24E+02
F22	5.92E+02	5.07E+02	5.68E+02	5.55E+02	6.31E+02
F23	8.80E+02	5.35E+02	8.27E+02	8.87E+02	9.04E+02
F24	2.14E+02	2.00E+02	2.68E+02	2.13E+02	2.19E+02
F25	2.09E+02	2.01E+02	2.01E+02	2.06E+02	2.05E+02

Table 16 Results for different mesh sizes for dimension 50

Test function	VMO(8)	VMO(12)	VMO(24)	VMO(50)	VMO(100)
F6	2.55E+03	1.78E+03	2.05E+03	2.59E+03	4.21E+03
F7	4.49E+03	3.71E+03	4.03E+03	4.26E+03	5.26E+03
F8	2.11E+01	2.11E+01	2.12E+01	2.11E+01	2.15E+01
F9	5.21E+02	2.28E+02	9.24E+02	3.17E+02	9.65E+02
F10	1.54E+02	1.07E+01	1.22E+02	1.28E+02	1.33E+02
F11	7.96E+01	4.24E+01	8.26E+01	8.63E+01	1.09E+02
F12	9.92E+04	9.67E+04	9.97E+04	1.42E+05	1.14E+06
F13	8.00E+00	8.03E+00	9.41E+00	8.16E+00	9.32E+00
F14	2.19E+01	2.19E+01	2.15E+01	2.11E+01	2.19E+01
F15	5.81E+02	4.26E+02	5.81E+02	5.90E+02	6.96E+02
F16	3.04E+02	2.00E+02	2.49E+02	3.56E+02	5.66E+02
F17	4.95E+02	2.72E+02	2.18E+02	5.25E+02	6.02E+02
F18	1.30E+03	9.26E+02	9.53E+02	1.14E+03	1.20E+03
F19	9.91E+02	9.41E+02	9.56E+02	1.07E+03	1.17E+03
F20	1.05E+03	9.29E+02	9.55E+02	1.03E+03	1.18E+03
F21	1.24E+03	9.60E+02	9.77E+02	9.96E+02	1.29E+03
F22	9.56E+02	9.83E+02	1.00E+03	9.98E+02	1.30E+03
F23	1.15E+03	1.02E+03	1.75E+03	1.57E+03	1.27E+03
F24	5.05E+02	2.00E+02	2.12E+02	1.10E+03	1.36E+03
F25	1.64E+03	1.68E+03	1.74E+03	1.71E+03	1.81E+03

Table 17 Results for different kinds of clearing operators for dimension 10

F	VMO-NC	VMO-AC	VMO-C1	VMO-C2	VMO-C3	VMO-C4	VMO-C5
F6	1.11E+08	6.02E+01	1.25E+03	3.83E+02	1.27E+02	1.76E+02	6.52E+02
F7	3.76E+03	6.39E+02	7.91E+02	5.71E+03	4.03E+03	1.57E+03	5.02E+03
F8	2.04E+01	2.03E+01	2.04E+01	2.04E+01	2.03E+01	2.04E+01	2.04E+01
F9	1.96E+03	3.10E+00	6.45E+01	3.85E+01	2.01E+01	1.27E+01	9.07E+00
F10	8.05E+01	4.79E+00	1.93E+01	1.09E+01	1.95E+01	5.08E+01	4.88E+01
F11	9.19E+00	5.12E+00	6.39E+00	3.77E+00	2.07E+00	4.36E+00	6.27E+00
F12	4.03E+04	7.55E+01	1.29E+03	5.09E+02	5.19E+02	3.83E+02	6.39E+02
F13	7.24E+00	7.48E-01	1.93E+00	1.83E+00	1.42E+00	1.93E+00	2.96E+00
F14	3.87E+00	2.74E+00	3.08E+00	3.17E+00	3.27E+00	3.55E+00	3.71E+00
F15	5.70E+02	2.14E+02	3.42E+02	2.85E+02	3.88E+02	3.56E+02	4.51E+02
F16	2.96E+02	1.03E+02	1.44E+02	1.19E+02	1.24E+02	2.05E+02	2.24E+02
F17	1.53E+02	1.20E+02	1.50E+02	1.34E+02	1.41E+02	2.16E+02	2.36E+02
F18	1.08E+03	7.46E+02	8.87E+02	8.72E+02	8.83E+02	1.02E+03	1.07E+03
F19	1.09E+03	6.98E+02	8.74E+02	8.40E+02	9.02E+02	1.03E+03	1.06E+03
F20	1.08E+03	7.87E+02	8.71E+02	8.50E+02	9.08E+02	1.04E+03	1.05E+03
F21	1.27E+03	5.53E+02	8.87E+02	7.55E+02	8.22E+02	1.27E+03	1.19E+03
F22	9.58E+02	7.51E+02	8.37E+02	7.81E+02	7.75E+02	9.13E+02	9.57E+02
F23	1.27E+03	6.45E+02	9.40E+02	8.96E+02	8.78E+02	1.23E+03	1.19E+03
F24	4.18E+02	4.06E+02	4.34E+02	3.96E+02	3.66E+02	5.92E+02	6.94E+02
F25	4.29E+02	3.88E+02	4.09E+02	4.03E+02	4.67E+02	1.45E+03	1.68E+03

Table 18 Results for different kinds of clearing operators for dimension 30

F	VMO-NC	VMO-AC	VMO-C1	VMO-C2	VMO-C3	VMO-C4	VMO-C5
F6	1.02E+10	2.82E+02	2.90E+04	5.03E+03	1.52E+03	9.17E+03	1.93E+03
F7	2.76E+03	2.72E+02	1.17E+04	1.05E+03	1.83E+03	2.72E+03	3.77E+03
F8	2.10E+01	2.09E+01	2.10E+01	2.09E+01	2.10E+01	2.10E+01	2.10E+01
F9	1.82E+04	1.04E+02	3.10E+02	1.99E+02	1.42E+02	1.82E+02	7.48E+02
F10	4.97E+02	5.31E+00	1.09E+02	8.48E+01	1.38E+02	4.09E+02	4.12E+02
F11	3.99E+01	1.41E+01	2.79E+01	1.54E+01	1.59E+01	2.41E+01	2.90E+01
F12	1.14E+06	9.56E+03	4.07E+04	2.24E+04	1.29E+04	4.85E+04	6.68E+04
F13	1.15E+02	4.52E+00	1.37E+01	1.16E+01	9.60E+00	1.35E+01	2.49E+01
F14	1.37E+01	1.01E+01	1.23E+01	1.24E+01	1.26E+01	1.31E+01	1.32E+01
F15	8.28E+02	3.36E+02	4.30E+02	4.30E+02	5.45E+02	6.79E+02	6.66E+02
F16	6.36E+02	2.68E+02	2.65E+02	2.90E+02	2.40E+02	5.01E+02	5.25E+02
F17	3.38E+02	1.97E+02	2.86E+02	2.69E+02	2.79E+02	5.59E+02	6.53E+02
F18	1.17E+03	8.07E+02	8.29E+02	8.27E+02	8.29E+02	1.09E+03	1.17E+03
F19	1.18E+03	8.06E+02	8.29E+02	8.27E+02	8.26E+02	1.08E+03	1.16E+03
F20	1.17E+03	8.06E+02	8.29E+02	8.28E+02	8.36E+02	1.13E+03	1.17E+03
F21	1.34E+03	5.00E+02	8.41E+02	5.71E+02	8.58E+02	1.27E+03	1.32E+03
F22	1.34E+03	5.07E+02	5.17E+02	5.67E+02	1.01E+03	1.26E+03	1.26E+03
F23	2.53E+03	5.35E+02	8.93E+02	8.82E+02	8.12E+02	1.04E+03	1.34E+03
F24	1.93E+03	2.00E+02	5.26E+02	3.05E+02	5.74E+02	1.10E+03	1.05E+03
F25	1.41E+03	2.01E+02	5.92E+02	5.43E+02	5.89E+02	5.99E+02	5.86E+02

Table 19 Results for different kinds of clearing operators for dimension 50

F	VMO-NC	VMO-AC	VMO-C1	VMO-C2	VMO-C3	VMO-C4	VMO-C5
F6	3.18E+10	1.78E+03	7.85E+06	3.82E+03	3.13E+03	3.61E+03	2.68E+03
F7	3.96E+03	3.71E+03	4.54E+03	3.71E+03	3.91E+03	3.81E+03	3.71E+03
F8	2.14E+01	2.11E+01	2.11E+01	2.11E+01	2.13E+01	2.11E+01	2.11E+01
F9	1.75E+04	2.28E+02	6.66E+02	2.33E+02	5.83E+02	6.19E+02	4.49E+02
F10	6.64E+02	1.07E+01	2.41E+01	1.65E+01	3.28E+01	3.88E+01	4.48E+01
F11	9.71E+01	4.24E+01	5.87E+01	3.80E+01	6.47E+01	7.17E+01	6.37E+01
F12	1.57E+06	9.67E+04	4.25E+05	1.93E+05	5.87E+05	6.07E+05	7.39E+05
F13	1.07E+02	8.03E+00	3.91E+01	3.09E+01	1.00E+02	9.33E+01	1.08E+02
F14	1.44E+01	2.19E+01	2.25E+01	2.23E+01	2.28E+01	2.21E+01	2.26E+01
F15	1.11E+03	4.26E+02	5.55E+02	5.42E+02	5.35E+02	5.75E+02	5.87E+02
F16	7.60E+02	2.00E+02	5.08E+02	5.33E+02	5.12E+02	5.23E+02	5.00E+02
F17	5.19E+02	2.72E+02	5.04E+02	5.01E+02	5.50E+02	5.84E+02	5.93E+02
F18	1.35E+03	9.26E+02	1.11E+03	1.21E+03	1.49E+03	1.63E+03	1.26E+03
F19	1.31E+03	9.41E+02	1.11E+03	1.09E+03	1.51E+03	1.76E+03	1.24E+03
F20	1.31E+03	9.29E+02	1.25E+03	1.23E+04	1.18E+03	1.28E+03	1.25E+03
F21	1.41E+03	9.60E+02	1.22E+03	1.29E+03	1.23E+03	1.28E+03	1.37E+03
F22	1.56E+03	9.83E+02	1.34E+03	1.19E+03	1.30E+03	1.29E+03	1.40E+03
F23	1.46E+03	1.02E+03	2.00E+03	1.84E+03	1.41E+03	1.52E+03	1.32E+03
F24	1.25E+03	2.00E+02	1.32E+03	2.09E+02	1.42E+03	1.85E+03	1.44E+03
F25	2.13E+03	1.68E+03	1.80E+03	1.78E+03	1.75E+03	1.65E+03	1.81E+03

Table 20 Results for the frontier's operator for dimension 10

Test function	VMO-F	VMO-NF
F6	6.02E+01	7.64E+00
F7	6.39E+02	6.38E+02
F8	2.03E+01	2.03E+01
F9	3.10E+00	7.38E+00
F10	4.79E+00	3.91E+01
F11	5.12E+00	6.57E+00
F12	7.55E+01	5.26E+02
F13	7.48E-01	1.52E+00
F14	2.74E+00	3.88E+00
F15	2.14E+02	4.44E+02
F16	1.03E+02	2.56E+02
F17	1.20E+02	1.86E+02
F18	7.46E+02	1.20E+03
F19	6.98E+02	1.15E+03
F20	7.87E+02	1.15E+03
F21	5.53E+02	1.19E+03
F22	7.51E+02	9.94E+02
F23	6.45E+02	1.28E+03
F24	4.06E+02	1.31E+03
F25	3.88E+02	1.85E+03

Table 21 Results for the frontier's operator for dimension 30

Test function	VMO-F	VMO-NF
F6	2.82E+02	3.36E+02
F7	2.72E+03	2.72E+03
F8	2.09E+01	2.10E+01
F9	1.04E+02	1.48E+02
F10	5.31E+00	7.69E+01
F11	1.41E+01	3.21E+01
F12	1.05E+04	7.24E+03
F13	4.52E+00	3.49E+01
F14	1.01E+01	1.32E+01
F15	3.36E+02	1.04E+03
F16	4.68E+02	6.31E+02
F17	1.97E+02	7.11E+02
F18	8.07E+02	1.33E+03
F19	8.06E+02	1.37E+03
F20	8.06E+02	1.33E+03
F21	5.00E+02	1.40E+03
F22	5.07E+02	1.46E+03
F23	5.35E+02	1.41E+03
F24	2.00E+02	1.42E+03
F25	2.01E+02	1.86E+03

Table 22 Results for the frontier's operator for dimension 50

Test function	VMO-F	VMO-NF
F6	1.78E+03	1.14E+02
F7	3.71E+03	3.71E+03
F8	2.11E+01	2.12E+01
F9	2.28E+02	4.10E+02
F10	1.07E+01	7.46E+01
F11	4.24E+01	6.19E+01
F12	9.67E+04	9.56E+04
F13	8.03E+00	1.21E+02
F14	2.19E+01	2.36E+01
F15	4.26E+02	9.59E+02
F16	2.00E+02	7.42E+02
F17	2.72E+02	9.89E+02
F18	9.26E+02	1.32E+03
F19	9.41E+02	1.35E+03
F20	9.29E+02	1.34E+03
F21	9.60E+02	1.02E+03
F22	9.83E+02	1.49E+03
F23	1.02E+03	1.47E+03
F24	2.00E+02	1.52E+03
F25	1.68E+03	1.89E+03

Table 23 Results of the comparison with others algorithms for dimension 10

Test function	ODE	LDWPSO	SSGA	VMO
F6	1.59E−01	5.68E+04	3.65E+04	6.02E+01
F7	1.27E+03	3.02E+00	3.36E+04	6.39E+02
F8	2.04E+01	2.05E+01	3.21E+04	2.03E+01
F9	7.67E-09	2.98E+01	3.06E+04	3.10E+00
F10	1.37E+01	3.46E+01	2.99E+04	4.79E+00
F11	1.75E+00	7.19E+00	2.92E+04	5.12E+00
F12	2.13E+01	1.44E+03	2.62E+04	7.55E+01
F13	6.47E−01	2.47E+00	2.65E+04	7.48E−01
F14	3.24E+00	3.62E+00	2.63E+04	2.74E+00
F15	3.04E+02	4.28E+02	2.63E+04	2.14E+02
F16	1.09E+02	1.72E+02	2.63E+04	1.03E+02
F17	1.34E+02	2.03E+02	2.57E+04	1.20E+02
F18	4.88E+02	9.34E+02	2.58E+04	7.46E+02
F19	4.84E+02	9.26E+02	2.43E+04	6.98E+02
F20	5.08E+02	9.27E+02	2.44E+04	7.87E+02
F21	5.40E+02	1.02E+03	2.37E+04	5.53E+02
F22	7.70E+02	8.57E+02	2.42E+04	7.51E+02
F23	5.66E+02	1.11E+03	2.39E+04	6.45E+02
F24	2.00E+02	5.29E+02	2.32E+04	4.06E+02
F25	1.74E+03	5.06E+02	2.36E+04	3.88E+02

Table 24 Results of the comparison with others algorithms for dimension 30

Test function	ODE	LDWPSO	SSGA	VMO
F6	1.31E−05	1.58E+09	1.37E+05	2.82E+02
F7	4.70E+03	2.84E+02	1.36E+05	2.72E+02
F8	2.10E+01	2.10E+01	1.32E+05	2.09E+01
F9	3.98E−02	2.42E+02	1.33E+05	1.04E+02
F10	1.17E+02	2.91E+02	1.32E+05	5.31E+00
F11	3.09E+01	3.73E+01	1.28E+05	1.41E+01
F12	4.94E+04	2.94E+05	1.26E+05	9.56E+03
F13	2.61E+00	1.40E+02	1.29E+05	4.52E+00
F14	1.32E+01	1.36E+01	1.26E+05	1.01E+01
F15	3.48E+02	7.11E+02	1.27E+05	3.36E+02
F16	1.51E+02	4.36E+02	1.25E+05	2.68E+02
F17	2.01E+02	5.48E+02	1.27E+05	1.97E+02
F18	9.05E+02	9.06E+02	1.24E+05	8.07E+02
F19	9.05E+02	9.02E+02	1.21E+05	8.06E+02
F20	9.05E+02	9.05E+02	1.24E+05	8.06E+02
F21	5.00E+02	1.04E+03	1.20E+05	5.00E+02
F22	8.68E+02	7.15E+02	1.20E+05	5.07E+02
F23	5.34E+02	1.09E+03	1.19E+05	5.35E+02
F24	4.17E+02	2.36E+02	1.18E+05	2.00E+02
F25	1.63E+03	2.61E+02	1.21E+05	2.01E+02

Table 25 Results of the comparison with others algorithms for dimension 50

Test function	ODE	LDWPSO	SSGA	VMO
F6	1.21E+05	1.20E+10	2.52E+05	1.78E+03
F7	2.31E+06	8.29E+02	2.54E+05	3.71E+03
F8	4.79E+04	2.11E+02	2.49E+05	2.11E+01
F9	9.05E-02	5.10E+02	2.40E+05	2.28E+02
F10	1.24E+01	2.45E+02	2.42E+05	1.07E+01
F11	3.32E+05	6.21E+02	2.35E+05	4.24E+01
F12	3.34E+09	2.15E+06	2.33E+05	9.67E+04
F13	3.14E+06	2.40E+03	2.33E+05	8.03E+00
F14	3.13E+02	3.96E+02	2.33E+05	2.19E+01
F15	2.25E+02	7.81E+02	2.31E+05	4.26E+02
F16	5.36E+05	5.27E+02	2.31E+05	2.00E+02
F17	2.15E+06	7.08E+02	2.30E+05	2.72E+02
F18	7.56E+05	1.04E+03	2.30E+05	9.26E+02
F19	7.01E+05	1.09E+03	2.28E+05	9.41E+02
F20	2.78E+06	7.17E+02	2.27E+05	9.29E+02
F21	5.55E+05	9.93E+02	2.24E+05	9.60E+02
F22	3.01E+05	5.41E+02	2.24E+05	9.83E+02
F23	4.81E+06	1.14E+03	2.23E+05	1.02E+03
F24	9.74E+05	1.16E+03	2.22E+05	2.00E+02
F25	3.90E+07	5.41E+02	2.19E+05	1.68E+03

6 Conclusions

A PMH-denominated variable mesh optimization (VMO) was introduced in this paper. It includes new ways of exploring the search space:

- To use attraction towards the node with the best fitness (local extremes) on the neighbourhood of each mesh node, towards the node with the best fitness of the mesh (global extreme) and the frontiers of the mesh in the same algorithm iteration. For these exploration methods, the algorithm obtains a suitable balance between the exploitation and the exploration of the search space.
- To conduct an elitist initial population selection in each iteration, taking into account the quality and the separability between the nodes by means of a clearing operator. This mechanism introduces diversity in the population that facilitates a larger exploration of the solution space. In addition, the population contains the best representative of each zone of the explored search space.
- The clearing operator functions in an adaptive manner because it decreases the allowed distance between the nodes as the algorithm is conducted, depending on the extent of each interval. This operator causes the method to start with a high exploration level that decreases as the allowed distance during the running of the algorithm. It occurs in reverse proportion to the exploitation level of the method.

It has been shown that the proposed VMO algorithm presents a good scalability level presenting good results with dimensions 30 and 50.

The promising research line initiated with the present optimization framework based on VMO is worthy of further study. We will extend our investigation to test different mechanisms to create new nodes towards local and global extremes (for example, some operators presented in Herrera et al. 2003). Furthermore, we will study the clearing operator to regulate the diversity levels that it introduces in the mesh and to see its influence on the behaviour of the VMO algorithm.

Appendix: Results of the experiments

In this appendix, the results used for the statistical analysis for each study case are presented. Each algorithm is run 25 times for each test function, and the average error of the best found solution is computed. The function error value for a solution x is defined as $(f(x_j) - f(x_j^*))$, where x_j^* is the global optimum of the function. The captions in the tables detail which experiment it belongs to and the dimension of the test functions used (see Tables 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25).

References

Brest J, Boskovic B, Greiner S, Zumer V, Maucec MS (2007) Performance comparison of self-adaptive and adaptive differential evolution algorithms. Soft Comput 11(7):617–629

Deb K (2001) Self-adaptive genetic algorithms with simulated binary crossover. Evol Comput J 9(2):195–219

Engelbrecht A (2006) Fundamentals of computational swarm intelligence. Wiley, New York

Fernandes C, Rosa A (2001) A study of non-random matching and varying population size in genetic algorithm using a royal road function. In: Proceedings of IEEE congress on evolutionary computation. IEEE Press, Piscataway, New York, pp 60–66

García S, Fernández A, Luengo J, Herrera F (2009) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput Appl 13(10):959–977

García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC2005 special session on real parameter optimization. J Heuristics 15:617–644

Glover FW, Kochenberger GA (2003) Handbook of metaheuristics (International Series in Operations Research & Management Science). Springer, Berlin

Herrera F, Lozano M (eds) (2005) Special issue on real coded genetic algorithms: foundations, models and operators. Soft Comput 9:4

Herrera F, Lozano M, Verdegay J (1998) Tackling realcoded genetic algorithms: operators and tools for the behavioral analysis. Artif Intell Rev 12(4):265–319

Herrera F, Lozano M, Sánchez A (2003) A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study. Int J Intell Syst 18(3):309–338

Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6(2):65–70

Iman R, Davenport J (1980) Approximations of the critical region of the Friedman statistic. Commun Stat 18:571–595

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, pp 1942–1948

Laguna M, Martí R (2003) Scatter search. Methodology and implementation in C. Kluwer, Dordrecht

Lozano M, Herrera F, Molina D (eds) (2011) Special issue on scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems. Soft Comput

Michalewicz Z, Siarry P (2008) Special issue on adaptation of discrete metaheuristics to continuous optimization. In: Eur J Oper Res 185:1060–1061

Minetti G (2005) Uniform crossover in genetic algorithms. In: Proceedings of IEEE fifth international conference on intelligent systems design and applications, pp 350–355

Rahnamayan S, Tizhoosh H, Salama M (2008) Solving large scale optimization problems by opposition-based differential evolution. IEEE Trans Comput 7(10):1792–1804

Sheskin DJ (2007) Handbook of parametric and nonparametric statistical procedures. Chapman and Hall/CRC

Shi Y, Eberhart C (1998) A modified particle swarm optimizer. In: Proceedings of IEEE international conference on evolutionary computation, pp 69–73

Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11:341–359

Suganthan P, Hansen N, Liang J, Deb K, Chen YP, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University. http://www.ntu.edu.sg/home/EPNSugan/

Syswerda G (1989) Uniform crossover in genetic algorithms. In: Schaffer J (eds) Proceedings of third international conference on genetic algorithms. pp 2–9 Morgan Kaufmann, San Mateo

Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics 1:80–83