

# An Efficient Hierarchical Clustering Technique for Speciation

A. Pétrowski

Département Informatique, Institut National des Télécommunications  
9 rue Charles Fourier, 91011 Evry Cedex, France  
E-mail: ap@etna.int-evry.fr, URL: http://etna.int-evry.fr/~ap

**Abstract** — The sharing method or the basic clearing procedure are reliable niching methods. However, they suffer from a quite high complexity: from  $\mathcal{O}(n^2)$  to  $\mathcal{O}(cn)$ , where  $c$  is the number of subpopulations and  $n$  is the population size. In this paper, an efficient hierarchical clustering technique is presented, reducing the complexity of these niching methods to  $\mathcal{O}(n \log n)$ . On the other hand, the minimal niche radius has to be estimated by the basic sharing method or the clearing procedure. However, this data is not easily available when the properties of the fitness function are *a priori* unknown. To solve this problem, a clustering algorithm that determines subpopulations from the fitness landscape, without requiring problem dependant parameters, was used. Experimental results for two easy and two hard functions are presented. These problems are solved with a high reliability. Moreover, experiments indicate that not only does a hierarchical clustering algorithm reduce the complexity of the considered niching method, but also that it reduces the required number of evaluations of the fitness function.

## I. GENETIC ALGORITHMS AND MULTIMODAL OPTIMIZATION

A simple genetic algorithm [1] (SGA) is suitable to search the optimum of unimodal functions in a bounded search space. However, both experiments and analysis show that the SGA is unable to find the multiple global maxima of a multimodal function [1, 2]. This limitation can be overcome by a mechanism that creates and maintains several subpopulations within the search space. As such, that each highest maximum of the multimodal function can at least attract one of them. These mechanisms are referred to as “niching methods” [2].

The basic sharing method [3], or the basic clearing procedure [4, 5], define a subpopulation as contained in a hypersphere of the search space, the radius of which is given as a parameter: the niche radius, identical to all hyperspheres. Therefore, the niche radius must be lower than, or equal to the smallest distance that exists between two desirable maxima of the fitness function. This parameter is difficult to estimate. Also, several subpopulations can be associated to a same peak. In this case, the population size, required to solve the optimization problem, should be increased. A better partition of the population should take the fitness landscape into account so that every desirable peak is covered by a minimum of subpopulations: ideally, only one per peak. The knowledge of parameter values that strongly depend on the fitness

function as the “niche radius” is not required when using the fitness landscape to find such a partition.

The complexity of the basic sharing method is  $\mathcal{O}(n^2)$ , where  $n$  is the population size. Yin and Germary have proposed a sharing scheme using cluster analysis to partition a population into subpopulations [6]. This technique offers an adaptive niche radius between values  $d_{min}$  and  $d_{max}$  which should be determined with several tests. Moreover, it has the advantage of a complexity in  $\mathcal{O}(cn)$ , where  $c$  is the number of subpopulations. This complexity is identical to the complexity of the basic clearing procedure.

This paper proposes a hierarchical clustering technique which uses the fitness landscape to build subpopulations. Its complexity is  $\mathcal{O}(n \log n)$ . Once the sharing method and the clearing procedure is defined in sections II and III, section IV proposes a fast heuristic that estimates the smallest distance between each peak of the fitness function and all the others. This heuristic can be directly used to set a niche radius specific to each subpopulation. In this way, the niching algorithms can be reused with few alterations, but the complexity remains  $\mathcal{O}(cn)$ . Section V introduces a hierarchical clustering algorithm, derived from the LBG vector quantization algorithm [9], so that the complexity becomes  $\mathcal{O}(n \log n)$ . The heuristic defined in section IV can then be used as a criterion to partition a population by the clustering algorithm. Experimental results are presented in section VI. They aim at showing the effect of clustering on populations, and at comparing different variants of the clearing algorithm.

## II. THE SHARING METHOD

The sharing method [3] is one of the most reliable niching methods. With a proportionate selection operator, the expected number of offspring of an individual  $i$  is proportional to its *shared fitness*  $\tilde{f}_i(g)$  at generation  $g$ . This depends on its fitness  $f_i(g)$  using the following expression:

$$\tilde{f}_i(g) = \frac{f_i(g)}{\sum_{j=1}^n sh(d_{ij}(g))} \quad (1)$$

with

$$sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}}\right)^\alpha, & \text{if } d < \sigma_{share} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

- $sh$  : sharing function
- $d_{ij}$  : genotypic or phenotypic distance between individuals  $i$  and  $j$
- $\sigma_{share}$  : threshold of dissimilarity (niche radius)
- $\alpha$  : constant (typically set to 1)

For the sharing method to work properly, the value of the niche radius  $\sigma_{share}$  should be lower than, or equal to the smallest distance between any pair of global maxima though this is not an absolute requirement [9]. Its algorithmic complexity is  $\mathcal{O}(n^2)$  [3].

### III. CLEARING THE SEARCH SPACE

#### A. Principles

Like the sharing method, the clearing procedure [4] is applied after evaluating the fitness of individuals and before applying the selection operator. The clearing algorithm uses a dissimilarity measure between individuals to determine if they belong to the same subpopulation, or not. This value could be the Hamming distance for binary coded genotypes, or the Euclidian distance for “real coded” genotypes. It could also be defined at the phenotype level.

Each subpopulation contains a dominant individual: the one that has the best fitness. The basic clearing algorithm preserves the fitness of the dominant individual while it resets the fitness of all the other individuals of the same subpopulation to zero. Thus, the clearing procedure fully attributes the whole resource of a niche to a single individual: the *winner*. The winner takes all resources rather than sharing them with the other individuals of the same niche, as is done in the sharing method.

A subpopulation can be defined with a niche radius similar to the one used in the basic sharing method [4]. Thus, if an individual belongs to a given subpopulation, its dissimilarity with the dominant individual is lower than a given threshold  $\sigma_{clear}$ : the *clearing radius*.

It is also possible to generalize the clearing algorithm by accepting several winners chosen among the best individuals of each niche. The capacity of a niche is defined as the maximum number of winners that this niche can accept. Notice that if a capacity greater than 1 is chosen, the set of winners for a given population is generally not unique. There is at least one reason to want capacities greater than one: Choosing capacities between one and the population size offers intermediate situations between the maximum clearing and a standard GA. If the capacities are equal to the population size, the clearing effect disappears and the search method becomes a standard GA.

A simplified version of the clearing procedure in pseudo code is given in [4]. It can be shown that the complexity of the basic clearing procedure is  $\mathcal{O}(cn)$ , where  $c$  is the

number of subpopulations and  $n$  is the population size [4]. The subpopulation size for each niche is similar to the one obtained by the sharing method for a given population size. Note that the clearing procedure naturally protects the good individuals from the destructive effects of the genetic drift.

#### B. An elitist strategy for the clearing procedure

An elitist strategy memorizes the best individual(s) of a population found before the application of genetic operators and passes it (them) unaltered on to the next generation. K.A. De Jong noted that this strategy can improve the performance of a GA for a unimodal fitness function. He also stated that the performance is degraded for the F5 function, for example, which is multimodal [7]. An important reason of this low performance lies in the increased premature convergence hazards, during which a large number of individuals concentrate on some maxima of the fitness function.

One way to avoid this problem is to limit individual density at every region of the search space. This is precisely one of the effects of niching methods. Then, the problem is to determine the best individuals of every subpopulation in order to preserve them. Now, the clearing procedure naturally supplies these individuals: they are the winners. If the preservation of all the winners immobilizes too great a fraction of the population to achieve good convergence properties, it is possible to use a more restrictive choice criterion, such as, for example, preserving only the winners with a fitness greater than the average before clearing. This is the method chosen for the experiments described below. Another possibility is to memorize only the dominant individual of each subpopulation.

An elitist version of the clearing procedure annihilates the destructive effects of the reproduction operators. As a result, the clearing procedure needs much smaller population sizes than the sharing method for a given problem [4]. This dramatically reduces the number of fitness evaluations required to solve a problem.

### IV. AN ADAPTIVE NICHE RADIUS

The basic clearing procedure or the sharing method requires that the user has an *a priori* idea of the niche radius. This value must be lower than or equal to the smallest distance between any pair of maxima. To get an estimate of it, it would be possible to use the method proposed by Goldberg and Richardson [3] when these maxima are equidistant. However, to estimate this parameter, it is necessary to know problem dependent data.

Yin and Gernay [6], using a sharing scheme, described a method to adapt the characteristics of each niche according to the fitness landscape through cluster analysis. Each accumulation of individuals is likely to correspond to a peak of the fitness function. Thus, such an accumulation is considered as belonging to a same subpopulation. However, this clustering algorithm does not take into account the fitness values. But these values give an important information to separate peaks. Some preliminary experiments, analog to those presented in the following,

showed that useful subpopulations could disappear with such a clustering method. So, individuals run quite a high risk of being misclassified if the fitness information is neglected.

This section presents a heuristic that refines the membership criterion of a subpopulation by taking into account the fitness of the individuals. It gives an estimate for a niche radius suited to each subpopulation according to the local fitness landscape. As the radius is determined for each subpopulation, the clearing procedure, or the sharing scheme of Yin and Gerny can be used with this heuristic without alteration. The complexity is still  $\mathcal{O}(cn)$ .

### The Heuristic

The heuristic presented here works when the global maxima constitute a finite set of points. It is based on the hypothesis that sufficient individuals exist in the neighborhood of every peak of the fitness function. Hence, the location and the widths of these peaks are correctly estimated. This may involve a population size greater than the one required with a fixed radius niching method. In this last case indeed, only one individual per peak or per niche is sufficient for the GA to work.

This heuristic consists in inscribing an hypersphere within each niche explored by at least one individual at least. Its radius  $\sigma_c$  is lower than or equal to the distance between the dominant individual of the niche and the nearest individual that is unlikely to belong to this niche. It is now necessary that this individual is not located on another peak, otherwise the niche radius would be too large and useful niches might be removed. A firm decision is not obtainable because the fitness function is partially known for the individuals that have been evaluated in the current and previous generations. But this situation should be rare in order that the niching method can work. On the other hand, associating more than one niche to a same peak, due to a too small estimate of the radius, will not be destructive under the condition that the population size is large enough.

Let  $d_c(i)$  be the distance between the dominant of subpopulation  $c$  and an individual  $i$ . The chosen heuristic is based on the study of a function  $\mathbf{r}(x)$  defined as follows. Let  $\mathcal{D}$  be the set of  $q+1$  real numbers  $\mathcal{D} = \{0, d_1, d_2, \dots, d_q\}$ . Let  $k$  be an integer, such as  $d_c(i) \in [d_k, d_{k+1})$ , then

$$\mathbf{r}(x) = \max_{\forall i \text{ such that } d_c(i) \in [d_k, d_{k+1})} f_i \quad (3-a)$$

otherwise

$$\mathbf{r}(x) = \mathbf{r}(d_{k-1}) \quad (3-b)$$

Choosing the maximum of values  $f_i$  as value of  $\mathbf{r}(x)$  ensures that the peaks already found are taken into account. Other functions have been tested, as the average of values  $f_i$  in each interval  $[d_k, d_{k+1})$ , but worst results were obtained.

The value of  $\sigma_c$  is lower than or equal to the smallest value of  $d_i$  such as  $\mathbf{r}(d_{i-1}) < \mathbf{r}(d_i)$  (figure 1):

$$\sigma_c \leq d_i, \forall i \text{ such that } \mathbf{r}(d_{i-1}) < \mathbf{r}(d_i) \quad (4)$$

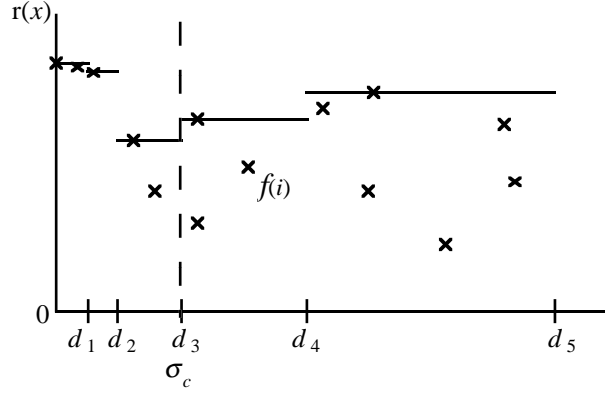


Fig. 1: calculation of radius  $\sigma_c$  for a set of points of coordinates  $(d_c(i), f_i)$ . These points are represented by symbols  $\times$ .

In the experiments, the sequence of values  $d_i$  is geometric to save memory and to ensure that the relative difference  $(d_i - d_{i+1}) / d_i$  is constant. The value of  $d_1$  has to be small enough to differentiate close peaks. In theory, it can be chosen arbitrarily small, but too small a value will lead to an unnecessary consumption of memory.

When the population size is not large enough, too few individuals may be outside the peaks. The risk is that it may not be possible to find a minimum for the function  $\mathbf{r}(x)$  between two peaks. It is possible to skirt the difficulty by considering all large values of  $\mathbf{r}(x)$ , greater than a threshold  $\tau_r$ , for  $x$  greater than a threshold  $\tau_\sigma$ , as the sign of the presence of a peak. This second criterion yields the following relation:

$$\sigma_c \leq d_i \quad \forall i \text{ such that } d_i > \tau_\sigma \wedge \mathbf{r}(d_i) > \tau_r \quad (5)$$

The maximal value that  $\sigma_c$  can take is chosen to guarantee a large enough diversity in the population regardless of the generation. If this maximal value  $\sigma_{\max}$  was not limited, the whole search space would be covered by only one niche, from which frequent premature convergences would occur. Choosing too small a maximal value would generate more niches than needed, and would require an increase of the population size. However, this value does not seem critical, and problem dependent knowledge is not required to determine it. In all the experiments described below, the maximal value of  $\sigma_c$  is chosen in such a way that the search space can be covered by a minimum of ten niches.

$$\sigma_c \leq \sigma_{\max} \quad (6)$$

$\sigma_c$  is taken as the greatest value satisfying the three relations (4), (5) and (6).

## V. REDUCING THE COMPLEXITY

This section presents a hierarchical clustering technique that distributes a population into subpopulations, allowing to use the clearing procedure or the sharing method. The concept of niche radius is now considered useless. The clustering technique performs a hierarchical vector quantization in the genotypic or phenotypic space. It adapts

the characteristics of the niches to the fitness landscape with the heuristic described in the previous section. Its algorithmic complexity is  $\mathcal{O}(n \log n)$ , while the basic clearing procedure or the sharing scheme proposed by Yin and Gerday have a complexity as  $\mathcal{O}(c n)$ , where  $c$  is the number of subpopulations, and  $n$  is the size of the population.

#### A. A Hierarchical Clustering Algorithm

The hierarchical clustering algorithm which was used is the standard "Tree-Structured Vector Quantization" (TSVQ) algorithm [12]. The algorithm is recursively defined on several levels. For each level, the LBG vector quantization algorithm [8] is applied to a set of points  $\mathbf{S}$  defined in a  $D$ -dimension space.  $\mathbf{S}$  is partitioned in  $k$  subsets  $\mathbf{S}_j, j \in [1, k]$ . The LBG algorithm is similar to the  $k$ -means algorithm. Let  $s_j$  be a point located at the centroid of the points of  $\mathbf{S}_j$ .  $s_j$  represents  $\mathbf{S}_j$ . A point  $x$  belongs to a subset  $\mathbf{S}_j$  if  $d(x, s_j) < d(x, s_i)$  for all  $j \neq i$ , where  $d(x, y)$  is the distance between the points  $x$  and  $y$ . The partition of a set of points  $\mathbf{S}$  respects the algorithm below:

Let  $n_j$  be the number of points belonging to the subset  $\mathbf{S}_j$ .

```

random initialization of  $k$  vectors  $s_j$ 
do
  initialize  $k$  vectors  $v_j$  to 0
  for all points  $x \in \mathbf{S}$ 
    select the integer  $e$  such that
       $d(x, s_e) < d(x, s_j)$  for all  $j \neq e$ ,
      where  $s_e$  is the centroid of  $\mathbf{S}_e$ 
     $v_e := v_e + x$ 
     $n_e := n_e + 1$ 
  end for
  for  $j = 1$  to  $k$ 
     $w_j := s_j$ 
     $s_j := v_j / n_j$ 
  end for
until  $s_j = w_j, \forall j \in [1, k]$ 

```

When the convergence is achieved, the set  $\mathbf{S}$  is partitioned into  $k$  subsets  $\mathbf{S}_j$ .

The partitioning procedure can be applied recursively on each subset obtained, until they contain only one point or are empty. Thus, a subset  $\mathbf{S}_i = \mathbf{T}$  can be partitioned into  $k$  subsets noted  $\mathbf{S}_{ij} = \mathbf{T}_j$  with  $1 \leq j \leq k$ . If a node of a graph is associated to each subset  $\mathbf{S}_i$ , an edge ties this node with those associated to the subsets  $\mathbf{S}_{ij}$ . This graph is a tree of root  $\mathbf{P}$ . Each leaf is either associated to a single point of  $\mathbf{P}$ , or to the empty set. At level  $l$ , a subset of  $\mathbf{P}$  is characterized by  $l$  subscripts  $\mathbf{P}_{i1, i2, \dots, il}$ .

#### Complexity

To each level, a set is partitioned into  $k$  subsets. So, if  $n$  is the number of points of  $\mathbf{P}$ ,  $\log_k n$  levels are required to classify every point. At level 1,  $\mathbf{P}$  is partitioned into  $k$  subsets  $\mathbf{P}_i$ . At level 2,  $\mathbf{P}$  will be partitioned into  $k^2$  subsets  $\mathbf{P}_{ij}$  and so on... On the other hand, at any level  $l$ ,  $k$

comparisons of distances must be performed to select the representative vector of a given point of a subset  $\mathbf{P}_{i1, i2, \dots, il}$ . This operation has to be repeated for all points of  $\mathbf{P}$ . Therefore,  $n k$  distance computations are needed for one level. In addition,  $n$  vector operations are performed to compute  $v_e$ . A maximum of  $n$  operations is required to obtain  $s_j$  from  $v_j$ .

The convergence at a given level *a priori* requires an indefinite number of iterations. However, some preliminary experiments have shown that the convergence criterion can be relaxed by adopting a number of iterations  $\phi(k)$  which only depends on  $k$ . The complexity of a level is therefore  $\mathcal{O}(n \phi(k) k)$ . This yields a global complexity as  $\mathcal{O}(n \phi(k) k \log n)$  for the  $\log_k n$  levels.

Choosing the smallest possible value for  $k$  leads a saving of time. Hence the reason why  $k$  is set to the value 2. In this case, the experiments have shown that choosing a number of iterations  $\phi(k) = 2$  does not greatly perturb the quality of the results for the present applications. The mean complexity of the clustering algorithm then becomes  $\mathcal{O}(n \log n)$ .

#### B. Partitioning a Population

Each individual is represented by a point in the space of genotypes or phenotypes. These individuals are classified according to the TSVQ algorithm presented above. Let  $\mathbf{P}$  be the set of individuals. Finding all the individuals of a subpopulation dominated by an individual  $\Delta$  is the same thing as finding the subset of points  $\mathbf{P}_{j1, j2, \dots, jl}$ , located on a peak with  $\Delta \in \mathbf{P}_{j1, j2, \dots, jl}$  for the smallest level  $l$ . This membership criterion is defined with the function  $\mathbf{r}(x)$  which was used to compute an adaptive radius  $\sigma_c$  (see section IV). The level  $l$  is the smallest level, such that there is no integer  $i$  which satisfies relations (4) or (5). The individuals which belong to  $\mathbf{P}_{j1, j2, \dots, jl}$  then constitute a subpopulation.

The complexity of the individual membership search of the population is  $\mathcal{O}(n)$ . In effect, the search of a dominant requires  $k \log n$  steps, where  $n$  is the size of the population and  $k = 2$ , to reach the leaf associated with this dominant in the binary search tree. Let  $l$  be the level so that  $\mathbf{P}_{j1, j2, \dots, jl}$  is the subpopulation that contains the dominant. Let  $n_l$  the cardinal of  $\mathbf{P}_{j1, j2, \dots, jl}$ . Then,  $2 n_l$  distance computations between the dominant and the other individuals belonging to  $\mathbf{P}_{j1, j2, \dots, jl-1}$  are required to establish that an individual of  $\mathbf{P}_{j1, j2, \dots, jl-1}$  satisfies at least relations (4) or (5).  $(4 n_l - 1)$  exploration steps in the classification tree are needed to identify the  $2 n_l$  individuals of  $\mathbf{P}_{j1, j2, \dots, jl-1}$ . So,  $K n$  operations are needed to compute function  $\mathbf{r}$ , where  $K$  is a positive constant. The total number of operations is the sum of the number of operations required to find the dominant, and the number of operations required to compute function  $\mathbf{r}$ . The contribution of the former is  $\mathcal{O}(\log n)$ , and the contribution of the latter is  $\mathcal{O}(n)$ . The logarithm is therefore neglected in the complexity calculation, which is now  $\mathcal{O}(n)$ .

Consequently, the complexity of the whole niching method is the complexity of the clustering algorithm, i.e.  $\mathcal{O}(n \log n)$ .

## VI. EXPERIMENTS

Several experiments have been performed in order to evaluate the robustness of the clustering method. Both easy and hard fitness functions have been considered. The experiments described here concern four functions. The "Roots" function takes its maxima on the sixth roots of the unity in the complex plane. The "M6" function [2] exhibits 25 maxima placed as a  $5 \times 5$  array and has only one global maxima. These problems are considered as quite easy. The two most difficult functions are the "M7" [9, 2] and "M9" [10, 2] massively multimodal deceptive function.

### A. The tested functions

#### The "M6" function

M6 is derived from the "F5" function used by De Jong in his dissertation [7]. It is the easiest function studied here. It shows 25 maxima located as though on a 2-dimensional array with values ranging from 476.191 to 499.002. The purpose of the present experiments was not only to find the global maximum located at  $(-32, -32)$ , but also to locate all the maxima.

"M6" is defined as follows:

$$M6(x, y) = 500 - \frac{1}{0.002 + \sum_{i=0}^{24} \frac{1}{1 + i + (x - X_i)^6 + (y - Y_i)^6}}$$

where  $(X_i, Y_i)$  is the location of the  $i$ -th maximum :

$$(X_i, Y_i) \in \{ \begin{aligned} &(-32, -32), (-32, -16), (-32, 0), (-32, 16), (-32, 32), \\ &(-16, -32), (-16, -16), (-16, 0), (-16, 16), (-16, 32), \\ &(0, -32), (0, -16), (0, 0), (0, 16), (0, 32), \\ &(16, -32), (16, -16), (16, 0), (16, 16), (16, 32), \\ &(32, -32), (32, -16), (32, 0), (32, 16), (32, 32) \end{aligned} \}$$

In the search space,  $x \in [-64, 64]$ , and  $y \in [-64, 64]$ .  $x$  and  $y$  are each encoded on 20 bits. Thus, the genotype has a length of 40 bits.

#### The "Roots" function

The "Roots" function is defined as follows:

$$\text{Roots}(z) = \frac{1}{1 + |z^6 - 1|}$$

where  $z \in \mathbf{C}$ .

This function takes its maxima at the sixth roots of unity in the complex plane.

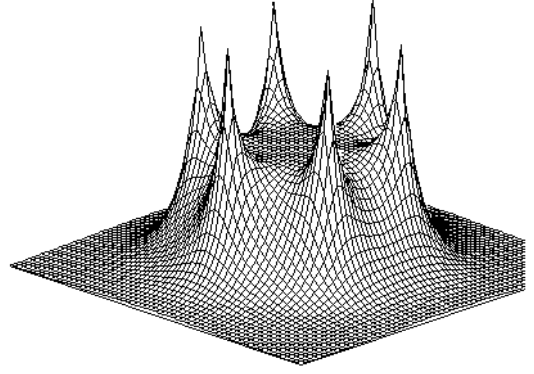


Fig. 2: the "Roots" function

This function presents a large plateau at the height of 0.5, centered at  $(0, 0)$  and surrounded by six thin peaks at the height of 1.0 (figure 2). The maxima are quite easy to find but the experiments lead one to think that this problem seems to be harder than "M6".

Let  $z = x + iy$ . In the search space,  $x \in [-2, 2]$ , and  $y \in [-2, 2]$ .  $x$  and  $y$  are each encoded on 20 bits. Thus, the genotype has a length of 40 bits.

#### The "M7" function

M7 is the name given by Mahfoud [2] to the massively multimodal deceptive function elaborated by Goldberg, Deb and Horn [9]. M7 is defined as follows:

$$M7(x_0, \dots, x_{29}) = \sum_{i=0}^4 \mathbf{u} \left( \sum_{j=0}^5 x_{6i+j} \right)$$

where  $\forall k, x_k \in \{0, 1\}$ . Function  $\mathbf{u}(x)$  is defined for the integer values 0 to 6 (figure 3). It has two maxima of value 1 at the points  $x = 0$  and  $x = 6$ , as well as a local maximum of value 0.640576 for  $x = 3$ . Function  $\mathbf{u}$  was specially built to be deceptive.

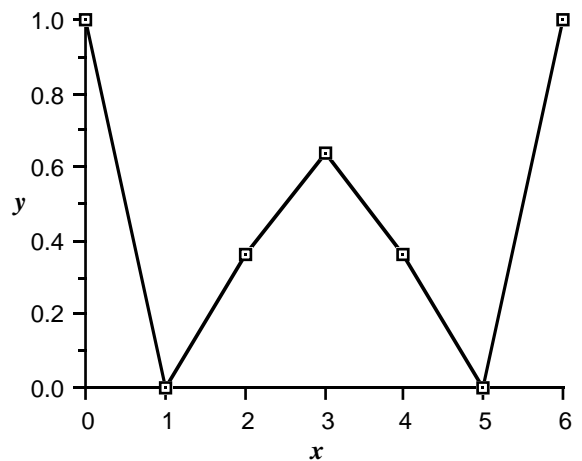


Fig. 3:  $y = u(x)$

Function M7 has 32 global maxima of value equal to 5, and several million local maxima, the values of which are between 3.203 and 4.641.

The concatenation of bits  $x_k$  directly constitutes a 30-bit genotype.

### The “M9” function

**M9** is the name given by Mahfoud [2] to the massively multimodal deceptive function elaborated by Horn and Goldberg [10]. “**M9**” is the sum of three elementary subfunctions  $f_{mdG}$  defined for 8-bit binary vectors:

$$M9(x_0, \dots, x_{23}) = \sum_{i=0}^2 f_{mdG}(x_{8i}, \dots, x_{8i+7})$$

where  $x_i \in \{0, 1\}$

Let  $\mathbf{G}$  be the set of the maxima of  $f_{mdG}$ . Let  $s$  be an 8-bit binary vector.

$$f_{mdG}(s) = \begin{cases} 10 & \text{if } s \in \mathbf{G} \\ \min_{g \in \mathbf{G}} H(s, g) & \text{otherwise} \end{cases}$$

where  $H(s, g)$  is the Hamming distance between  $s$  and  $g$ .  $\mathbf{G}$  contains three arbitrarily chosen points: 00000000, 10001100, 01001010.

“M9” is hard because the peaks are isolated and located at the lower points of large basins. To illustrate this, one can consider a generalization of the  $f_{mdG}(s)$  function for real numbers, by replacing the Hamming distance by the Euclidian distance. Figure 4 illustrates of such a function in the monodimensional case.

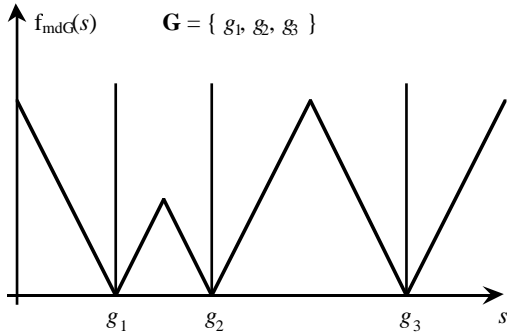


Fig. 4:  $f_{mdG}(s)$  function in the monodimensional real case.

### B. Experimental conditions

The only niching method used for the experiments was the clearing procedure only. In effect, the sharing method was unable to find all the global maxima of “M7” and “M9” [2]. Moreover, in the case of easy functions such as “M6” and “Roots” the sharing method required much greater population sizes than the clearing procedure. These had to be of the order of 1000 individuals instead of 100, and the tests proved very slow.

To ensure that the hierarchical clustering technique, associated to the “adaptive niche radius” heuristic, is interesting, two quality requirements have to be reached: the premature convergence rate and the mean number of evaluations of the fitness function to obtain all the global maxima. These quantities has to be lower than or equal to the results given by the basic clearing procedure. These

quality criteria are a function of the population size. As part of the experiments, the performance of the “adaptive niche radius” heuristic, used without clustering analysis (see the end of section IV), was also measured.

First of all, the experiments aimed at finding the minimal population sizes that satisfy the first quality criteria, i.e. a number of premature convergence equal to zero over 100 successive tests for the four considered functions. It is considered that a premature convergence occurs when all the global maxima cannot be obtained because of allele losses. A test consists in evolving a population over 100 generations. The population sizes increase from 100 to 1000 individuals, by step of 100 individuals.

Secondly, once the minimal sizes of the populations was known, the computation time and the mean number of evaluations, required to find all the global maxima over 100 successive tests, were measured.

The parameters of the GA are given below:

- ◊ Binary coding of the genotype;
- ◊ Stochastic Universal Selection [11];
- ◊ One point crossover; The crossover rate is 1;
- ◊ The mutation rate is zero. Some diversity is generated in the population by replacing 10% of its individuals by random individuals at every generation;
- ◊ The minimal number of subpopulations obtained by the clustering algorithm is set to 16, in order to guarantee a large enough diversity;
- ◊ Capacity  $\kappa$  of the niches is set to 1 in order to guarantee a maximal diversity. This value is always adequate when searching a maximal diversity [4];
- ◊ The adaptive elitist strategy described above is used.

Concerning the “adaptive niche radius” technique (section IV), its maximal value is chosen in such a way that 16 niches at least can coexist. Thus,  $\sigma_{\max} = 0.25$  for the “Roots” and “M6” functions and  $\sigma_{\max} = 0.5$  for the “M7” and “M9” functions. Note that the distance unity is normalized, ensuring that the furthest points in the search space are at distance 1.

### C. Results

Figures 5 and 6 show real distributions examples of individuals among the subpopulation for the 2-dimension functions “M6” and “Roots” with the hierarchical clustering technique at generation 100. The points represents the phenotypes expressed in a 2-dimension space. The circled individuals belong to a same subpopulation. The function maxima are localized at the center of symbols “□”. The grey areas represent their associated subpopulations.

For the “Roots” function, notice that the partitioning creates many more subpopulation than needed. This is not surprising because the minimum number of subpopulations

was fixed on 16 to ensure a large enough diversity during the evolution of the population.

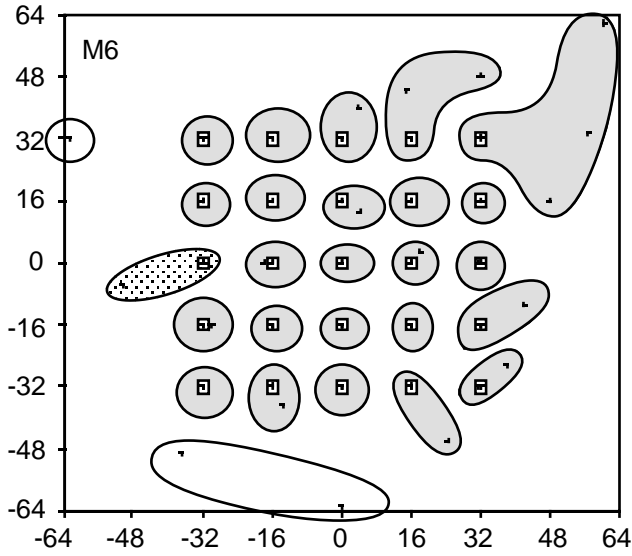


Fig. 5. subpopulations at generation 100 for the M6 function.

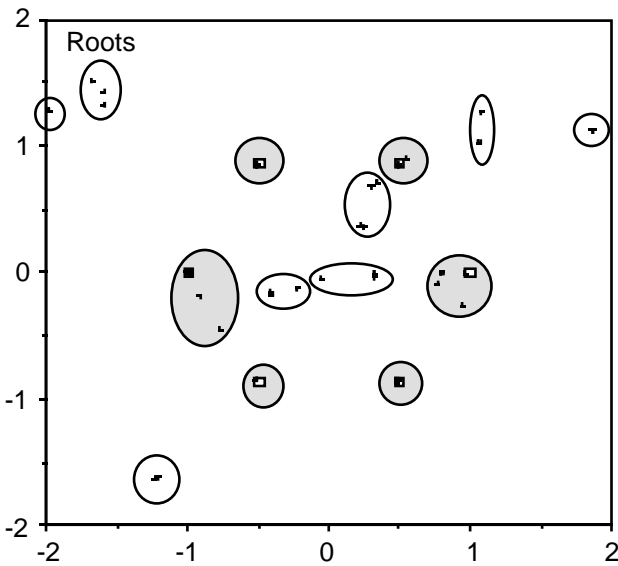


Fig. 6. subpopulations at generation 100 for the "Roots" function.

The tables below give the minimal population sizes, so that all the global peaks for the "Roots", "M7", and "M9" functions are discovered at every test. Concerning "M6", the problem was to find the 25 maxima of the function. 100 successive tests were performed using the following versions of the GA:

- ◊ the basic clearing procedure;
- ◊ the clearing procedure associated to an adaptive niche radius;
- ◊ the clearing associated to a hierarchical clustering technique.

The results of the measures are given on tables 1, 2 and 3:

**Table 1**

Basic Clearing Procedure	minimal population size	mean number of evaluations	duration of a generation
"M6"	100	$1.7 \cdot 10^3$	$4.3 \cdot 10^{-2}$ s
"Roots"	100	$2.4 \cdot 10^3$	$2.6 \cdot 10^{-2}$ s
"M7"	600	$2.2 \cdot 10^4$	4.3 s
"M9"	1000	$3.0 \cdot 10^4$	$1.3 \cdot 10^1$ s

**Table 2**

Adaptive Niche Radius	minimal population size	mean number of evaluations	duration of a generation
"M6"	100	$1.5 \cdot 10^3$	$4.1 \cdot 10^{-2}$ s
"Roots"	100	$2.2 \cdot 10^3$	$4.1 \cdot 10^{-2}$ s
"M7"	800	$1.5 \cdot 10^4$	$1.1 \cdot 10^1$ s
"M9"	> 1000 *	$1.5 \cdot 10^4$	$1.4 \cdot 10^1$ s

\* premature convergence sometimes occurs over 100 tests with 1000 individuals. The results are given for 1000 individuals.

**Table 3**

Hierarchical Clustering	minimal population size	mean number of evaluations	duration of a generation
"M6"	100	$1.0 \cdot 10^3$	$4.3 \cdot 10^{-2}$ s
"Roots"	100	$1.7 \cdot 10^3$	$4.2 \cdot 10^{-2}$ s
"M7"	800	$1.2 \cdot 10^4$	1.8 s
"M9"	1000	$1.4 \cdot 10^4$	1.9 s

The numbers of evaluations in the tables can be compared with those required by the sharing method or the deterministic crowding. These results, summarized in Table 4, are drawn from [2]. The population sizes were not given.

**Table 4**

number of evaluations	sharing method	deterministic crowding
"M6"	$1.3 \cdot 10^4$	$> 1.5 \cdot 10^6$
"M7"	$> 1.5 \cdot 10^6$	$1.0 \cdot 10^5$
"M9"	$> 1.5 \cdot 10^6$	$1.3 \cdot 10^6$

The adaptive techniques need fewer evaluations of the fitness function than the clearing procedure, implemented with a fixed niche radius. This is directly due to the effect of the adaptive elitist strategy (section III.B) that preserves more individuals, generations by generations, with adaptive niche radius techniques. The reasons of this behavior have not yet been analyzed. The niche radius techniques can require, on the other hand, at least for "M7", slightly greater population sizes to obtain a given premature convergence rate.

The speeded up clearing technique offers a real saving of time when population sizes are great, depending on the duration of the fitness evaluations. However, this saving of time is negligible compared to the time needed by the reproduction operators and the fitness evaluations for small population sizes.

## VII. CONCLUSION

This paper has presented a hierarchical clustering technique which allows the partition of a population into subpopulations, each of them being associated with a peak of the fitness function. Using this technique, the complexity of niching methods such as the sharing method or the clearing procedure was reduced to  $\mathcal{O}(n \log n)$ . The individuals located on a given peak of the fitness function were recognized with the use of a heuristic in a reliable way. Not only was the search space well covered by the subpopulations, but parameters difficult to estimate such as the niche radius were no longer necessary.

For reasons of efficiency, all the experiments were performed with the clearing procedure. This proved that the use of a clustering algorithm guaranteed the reliability of the basic niching algorithm, even for difficult functions. Not only is the number of the fitness function evaluations, required to solve the problems, is smaller than those of the basic clearing procedure, but this procedure also seems more efficient than the sharing or deterministic crowding techniques.

## REFERENCES

[1] D.E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley, 1989.

[2] S.W. Mahfoud, *Niching Methods for Genetic Algorithms*, doctoral dissertation, University of Illinois at Urbana-Champaign, 1995.

[3] D.E. Goldberg, J. Richardson, "Genetic Algorithms with Sharing for Multimodal Function Optimization", in *Genetic Algorithms and their Applications: Proc. of the 2nd Int. Conf. on Genetic Algorithms*, pp. 41-49, 1987.

[4] A. Petrowski, "A Clearing Procedure as a Niching Method for Genetic Algorithms", *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pp. 798-803, IEEE, 1996.

[5] A. Petrowski, "A New Selection Operator Dedicated to Speciation", in Th. Baeck, ed., *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA 97)*, Morgan Kaufmann, San Francisco, CA, 1997.

[6] X. Yin, N. Gernay, "A Fast Genetic Algorithm with Sharing Scheme using Cluster Analysis Methods in Multimodal Function Optimization", in R.F. Albrecht, C.R. Reeves & N.C. Steele (Eds.), *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, pp. 450-457, Berlin, Springer Verlag, 1993.

[7] K.A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, doctoral dissertation, University of Michigan, 1975.

[8] Y. Linde, A. Buzo, R.M. Gray "An Algorithm for Vector Quantization Design", *IEEE Trans. Commun.*, Vol. 28, pp. 84-95, Jan. 1980.

[9] D.E. Goldberg, K. Deb, J. Horn, "Massive Multimodality, Deception and Genetic Algorithms", in R. Männer, B. Manderick (Eds.) *Parallel Problem Solving from Nature 2*, pp. 37-46, North Holland, 1992.

[10] J. Horn, D.E. Goldberg, Genetic Algorithms Difficulty and the Modality of Fitness Landscapes, In L.D. Whitley & M.D. Vose (Eds.), *Foundations of Genetic Algorithms 3 (FOGA 3)*, pp. 243-269, Morgan Kaufmann, 1995.

[11] J.E. Baker, "Reducing Bias and Inefficiency in the Selection Algorithm", in *Genetic Algorithms and their Applications: Proc. of the 2nd Int. Conf. on Genetic Algorithms*, pp. 14-21, 1987.

[12] A. Buzo, A.H. Gray, R.M. Gray, J.D. Markel "Speech coding based upon vector quantization", *IEEE Trans. on Acoustic Speech and Signal Processing ASSP-28* pp.562-574. (October 1980).