A fuzzy clustering-based niching approach to multimodal function optimization

Action editor: Xin Yao


Laboratoire LCS, Faculté des Sciences, Université Mohammed V-Agdal, B.P. 1014, Av. Ibn Battouta, Rabat, Morocco

Received 20 June 1999; accepted 13 November 1999

Abstract

This paper presents a new method, which combines sharing and a fuzzy clustering technique to improve the performance of genetic algorithms in multimodal function optimization. This approach permits some limitations of the traditional sharing scheme to be overcome. Without using any prior information, it allows both location and maintenance of niches. Computer simulations show good performance for several multimodal test functions including a deceptive problem. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Genetic algorithms; Multimodal optimization; Sharing; Self learning; Fuzzy clustering

1. Introduction

A great deal of research has gone into finding efficient methods for locating the global optimum of a function. However, real optimization problems are often represented by multimodal functions and so require the identification of multiple optima, either global or local. Canonical genetic algorithms (GAs), used as general optimization techniques, always converge to a single solution and thus are not appropriate for multimodal optimization (Säreni and Krähenbühl, 1998). As a result, various mechanisms have been proposed, allowing GAs to identify multiple optima reliably. These methods work by encouraging artificial niche formation (Goldberg and Richardson, 1987; Mahfoud, 1992; Beasley et al., 1993a; Forrest et al., 1993; Harik, 1995; Miller and Shaw, 1995; Petrowski, 1996). However, most of them introduce one or more parameters that affect the algorithm performance, e.g., the sharing radius in fitness sharing or the crowding factor in crowding (Goldberg and Wang, 1997).

The purpose of this study is to design a new iterative technique for the formation and maintenance of stable sub-populations in GAs. To achieve this goal, a fuzzy clustering method is used in order to identify clusters that correspond to niches. In addition, for each detected cluster \(C_i\), the algorithm provides some important features, such as the cluster center \(y_i\) (optima), the minimum and maximum values of internal similarities \(S_{\text{min}}, S_{\text{max}}\), the cluster...
radius \((r_i)\) and size. To promote stable sub-populations, a spatial separation procedure is also retained so that search occurs at multiple exploration and exploitation levels. Furthermore, as the niching radii are dynamically adjusted, the proposed approach permits the location of non-detected niches and also provides a fine local tuning in order to obtain more accurate optima.

The proposed approach uses a three-layer strategy. The first layer is a GA with sharing (GASH), the second one is an unsupervised fuzzy clustering algorithm (FC), and the third layer implements the spatial separation principle (SS).

This paper starts with a brief description of different niching GAs (NGAs). The techniques involved in the different layers of the model, particularly the sharing and the fuzzy clustering methods, are then presented. Section 3 describes the structure of the underlying algorithm. Finally, in Section 4, several well-known functions are used to test the performance of the proposed algorithm. These functions have several characteristics which make them ideal for testing the ability of a GA to identify optima in a multimodal domain (Beasley et al., 1993a; Sareni and Krähenbühl, 1998).

2. GAs in multimodal function optimization

2.1. GA principle

GAs are computational models that emulate evolutionary behavior of biological systems to solve optimization problems (Goldberg, 1989). A GA comprises a set of individual elements (population) and a set of biologically inspired operators. According to the evolutionary theory, the population evolves towards increasingly better regions of the search space by means of selection, crossover and mutation processes.

A GA is generally initialized with a randomly generated population \(P(t = 0)\) of chromosomes. At each iteration \(t\), the selection process is applied to the current population \(P(t)\) to create an intermediate one. Pairs of parents are then randomly chosen for reproduction via a crossover procedure that mimics biological mating. Information between the two parents is exchanged and swapped to create two new offspring. To ensure that all points in the search space can be reached, the mutation operator is used to randomly alter the values of the new offspring by adding some small perturbations (Michalewicz, 1996). The process of going from the current population to the next one constitutes one generation in the execution of a GA. Furthermore, for preserving best strings, the elitist model is used as a replacement strategy, where the best chromosome is copied into the succeeding generation.

2.2. Niching methods

GAs have proven to be useful in several search and optimization problems. However, a simple GA always converges to a single optimum and is not suitable for multimodal optimization. To overcome this drawback, many theoretical and empirical studies have been realized. In difficult function optimization problems, such as those encountered with multimodal and deceptive functions, some investigations emphasize finding multiple solutions including local optima (Mahfoud, 1995). These techniques, called niching, attempt to preserve useful diversity in a GA, in order to slow down the GA’s premature convergence to a sub-optimal solution, and to permit the identification of multiple optima. The principle of these methods suggests that populations are actually divided into semi-isolated sub-populations or demes, each of which is relatively small. Two criteria are generally used in this context: (1) to maintain an appropriate number of individuals per separate deme by changing the selective environment of some sections of the search space so as to prevent the GA from quickly converging to those sections, and (2) to force local competition so as to limit reproduction between certain members of the population and hence allow the formation of ‘demes.’ The former criterion requires methods such as pre-selection, crowding and sharing. The latter requires approaches such as island and isolation models (Harik, 1994).

A number of niching mechanisms have been proposed in literature. One of the earliest was Cavicchio’s pre-selection (Mahfoud, 1992), in which offspring could only replace one of their parents. Boltzman tournament selection has also been shown to have niching effects (Goldberg, 1990).
Other important niching methods have been reported in the literature. The basic principles of these methods are presented below.

2.2.1. Crowding method

Crowding methods insert new elements in the population by overwriting similar elements. In the standard crowding, only a fraction of the global population (generation gap) reproduces and dies each generation. This scheme was unable to maintain more than two peaks on all problems. In the deterministic crowding, Mahfoud improved standard crowding by introducing competition between children and parents of identical niches. After crossover and eventually mutation, each child replaces the nearest parent if it has a higher fitness (Mahfoud, 1995).

2.2.2. Sequential niching

This method works by iterating a simple GA and maintaining the best solution of each run off-line. To avoid converging to the same area of the search space multiple times whenever a solution is located it depresses the fitness landscape at all points within some radii of that solution. This ensures that on subsequent runs the same peak will not be rediscovered. This technique has many similarities with fitness sharing. However, instead of the fitness of an individual being reduced because of its proximity to other members of population (sharing), individuals have their fitness reduced because their proximity to peaks located in previous runs (Beasley et al., 1993a). This method suffers similar problems with regards to choice of niche parameters (Beasley et al., 1993b).

2.2.3. Artificial immune systems

Immune systems have been studied by the artificial-life community both because of their intrinsic scientific interest and because of potential applications of ideas from immunology to computational problems (Mitchell and Forrest, 1993). Immune systems models have been applied to simulations in which a population of bit strings, called antibodies, evolves to match or cover a set of bit-string patterns, called antigens. The antigens define multiple spatial environments that allocate increasing fitnesses to population elements as they get closer to matching one or more antigens. The overall goal of this approach is to understand the real immune system and to use ideas generated from studying it to construct a powerful parallel algorithm (Forrest et al., 1993).

Forrest’s model works as follows: all the individuals initially had their fitness assignments set at zero. Then, an antigen and a sample of individuals are chosen at random. The individual best matching the antigen wins the competition and has its matching score added to its fitness (Smith et al., 1993). The effect of this process is similar to that of fitness sharing in the sense that highly similar individuals would often share the payment provided by any number of antigens that they matched particularly well.

The artificial immune model addresses an important problem of practical significance such as the network intrusion detection (Hofmeyr and Forrest, 1999). Though this work is not immediately applicable to function optimization it seems close in nature to an algorithm that might work in that domain (Harik, 1994).

2.2.4. Dynamic sharing

Dynamic niche sharing (Miller and Shaw, 1995) defines a fixed number of dynamic niches with radii and centers determined by a full population sort. For those individuals not in a niche, regular fixed sharing is used. This method requires an estimate of the number of peaks in addition to the niche radius. It presents certain advantages over other nicher, but the primary weakness of the scheme is the use of fixed sharing outside the dynamic niches (Goldberg and Wang, 1997).

2.2.5. Clearing procedure

The clearing technique is similar to fitness sharing. It introduces the concept of dominant individuals in the niche. Instead of evenly sharing the resources among the individuals of a sub-population as in fitness sharing, clearing attributes them only to the best members of each sub-population. A sub-population can be defined with a niche radius similar to the one used in the basic sharing method. Thus an individual belongs to a given sub-population only if its dissimilarity is less than a given clearing radius (Petrowski, 1996). Its reliability is similar to that of
the basic sharing method with lower complexity but, like sharing, it requires an a priori knowledge of the niche radius (Petrowski, 1997).

2.2.6. Coevolutionary sharing (CS)

The CS model (Goldberg and Wang, 1997), which attempts to directly avoid the estimation of niche radius, uses two coevolving populations: customers and businessmen. Customers are served by the nearest businessman. Using a sharing function, customer fitness is derated in proportion to the total number of other customers served by the nearest businessman. Thus, there is pressure to find businessmen serving relatively few customers. The customer population evolves under a traditional GA. In contrast, the businessmen attempt to maximize the number of customers served, having more customers yields higher fitness. To prevent convergence of the businessman population to a single global optimum, businessmen must be separated by a distance $d_{\text{min}}$. The businessman population evolves via a mechanism in which the best customers are converted into businessmen. For each businessman, $n$ customers are selected at random. The first customer that is both more fit and at $d_{\text{min}}$ away from other businessmen then replaces the original businessman in the population (Watson, 1999).

2.3. Fitness sharing

The sharing (Goldberg and Richardson, 1987) method is probably the most widely used among niching techniques (Săreni and Krăhenbühl, 1998). Fitness sharing modifies the landscape by reducing the payoff in densely populated regions. It lowers each population element’s fitness by an amount nearly equal to the number of similar individuals in the population. Typically, the shared fitness $f_{\text{sh}}(i)$ of an individual $i$ is its prior fitness $f(i)$ divided by its niche count:

$$f_{\text{sh}}(i) = \frac{f(i)}{\sum_{j=1}^{N} \text{sh}(d(i, j))}.$$  

(1)

An individual’s niche count is calculated by summing a sharing function over all members of the population. $N$ denotes the population size and $d(i, j)$ is a distance measure between the individuals $i$ and $j$. The sharing function (sh) measures the similarity level between two population elements. The most widely used function is:

$$\text{sh}(d) = \begin{cases} 1 - \frac{(d/s)^\alpha}{\sigma_s} & \text{if } d < \sigma_s \\ 0 & \text{otherwise} \end{cases}$$  

(2)

where $\sigma_s$ denotes a threshold of dissimilarity and $\alpha$ is a constant which regulates the shape of the sharing function. $d(i, j)$ is a distance measure between the two individuals $i$ and $j$. It is characterized by a similarity metric that can be based on either genotypic or phenotypic similarity. Genotypic similarity corresponds to bit-string representation and is generally the Hamming distance. Phenotypic similarity is directly linked to real parameters of the search space. It can be the Euclidean distance for instance. Sharing based on phenotypic similarity may give slightly better results than sharing with genotypic similarity (Deb and Goldberg, 1989).

There are several difficulties with the sharing approach. Generally, the number of expected optima and the niche radii are often estimated as the maximum number of peaks that could be in the domain, and the minimum niche radius of any optima within the domain, respectively (Miller and Shaw, 1995).

But, it may be difficult to accurately estimate the number of expected peaks in the domain. Furthermore, making parameter $\sigma_s$ the same for all individuals means that those peaks are considered as nearly equidistant in the domain (Săreni and Krăhenbühl, 1998). However, the peaks are often irregularly distributed through the space and they can vary significantly in length, width and shape. To overcome these limitations, our approach uses a fuzzy clustering technique in order to determine automatically the number of niches. Furthermore, the radius of each niche is continuously updated until an optimal solution is reached.

3. Fuzzy clustering

Cluster analysis attempts to organize a set of unlabeled input data into a number of natural groups (or clusters) in such a way that elements within a same cluster are as similar as possible while the
elements belonging to different clusters are highly dissimilar.

Clustering algorithms proposed in literature can be divided into two main categories: crisp (or hard) clustering procedures where each data point belongs to only one cluster, and fuzzy clustering techniques where every data point belongs to every cluster with a specific degree of membership (Duda and Hart, 1973).

Fuzzy clustering can be considered as a generalization of hard clustering and presents the advantage of dealing efficiently with overlapping clusters.

3.1. Definition of a fuzzy partition

Let $X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^p$ be a finite set of $n$ $p$-dimensional data points. A partition of $X$ into $C$ clusters can be defined with a matrix $U=\{\mu_{ij}\} \in \mathbb{R}^{n \times C}$, which satisfies the three conditions:

$$\sum_{j=1}^{C} \mu_{ij} = 1 \quad 1 \leq i \leq n \quad 1 \leq j \leq C, \quad (3a)$$

$$\sum_{i=1}^{n} \mu_{ij} = 1 \quad 1 \leq i \leq n, \quad (3b)$$

$$0 < \sum_{i=1}^{n} \mu_{ij} < n \quad 1 \leq j \leq C. \quad (3c)$$

The degree of membership $\mu_{ij}$ of $x_i$ to the $j$th cluster indicates the strength of assignation of $x_i$ to cluster $j$. The higher $\mu_{ij}$; the more reliable is the assignation of $x_i$ to this cluster.

Let $M_{nC}$ denotes the set of all matrices of $\mathbb{R}^{n \times C}$ satisfying Eqs. (3a), (3b) and (3c). The problem of partitioning $X$ into $C$ fuzzy subsets can be posed in terms of finding the matrix $U^* \in M_{nC}$, which is the best description of the cluster structure of $X$. The criterion used in this study, to evaluate the quality of the partition, is the normalized partition entropy defined as (Bezdek, 1981):

$$h(U) = -\frac{1}{\log(C)} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{C} \mu_{ij} \log(\mu_{ij}). \quad (4)$$

$h(U)$ can be interpreted as a measure of the fuzziness of the corresponding partition. The minimum value of $h$ ($h_{\text{min}}=0$) corresponds to a non-fuzzy (or hard) partition, where each object $x_i$ is assigned to a cluster $j^*$ such that:

$$\mu_{ij^*} = 1 \quad \text{and} \quad \mu_{ij} = 0 \quad \forall j \neq j^*. \quad (5a)$$

On the other hand, when the partition $U$ is the fuzziest one, i.e., when all objects have the same membership degree to all clusters

$$\mu_{ij^*} = \frac{1}{C}, \quad \forall i, j; 1 \leq i \leq n \quad \text{and} \quad 1 \leq j \leq C. \quad (5b)$$

$h(U)$ reaches its maximum value ($h_{\text{max}}=1$). Consequently, the closer the entropy is to zero, the better the fuzzy $C$ partition is.

3.2. The fuzzy C-means algorithm

Another useful bit of information about the cluster structure of $X$ can be given by its cluster centers $\nu=(\nu_1, \nu_2, \ldots, \nu_C)^T \in \mathbb{R}^{Cp}$ where $\nu_j$ is the $j$th cluster center, $\nu_j \in \mathbb{R}^p$, $1 \leq j \leq C$. The fuzzy $C$-means algorithm (Bezdek, 1981) is a partitioning procedure for optimizing the objective function $J_m$ defined by:

$$J_m(U, V) = \sum_{i=1}^{n} \sum_{j=1}^{C} (\mu_{ij})^m ||x_i - \nu_j||^2, \quad (7)$$

where $1 \leq m \leq \infty$ and $|| \cdot ||$ is any inner product norm on $\mathbb{R}^p$.

For $m > 1$, Bezdek proves that a minimizer $(U^*, V^*)$ of $J_m(U, V)$ must verify the two necessary conditions:

$$\nu_j^* = \frac{\sum_{i=1}^{n} (\mu_{ij})^m x_i}{\sum_{i=1}^{n} (\mu_{ij})^m}, \quad \forall j, 1 \leq j \leq C, \quad (8)$$

$$\mu_{ij}^* = \left[ \frac{\sum_{k=1}^{C} \left( \frac{|x_i - \nu_j|}{|x_i - \nu_k|} \right)^{2 \frac{m}{m-1}} }{\sum_{k=1}^{C} \left( \frac{|x_i - \nu_j|}{|x_i - \nu_k|} \right)^{2 \frac{m}{m-1}}} \right]^{-1}. \quad (9)$$

4. Description of the proposed approach (GASH-FC)

The principle of this new iterative technique is based on a three-layer strategy (Fig. 1). The first layer (GASH) is a GA, which combines sharing, and
mating restriction to avoid the formation of lethal individuals and to encourage speciation of a random initial population. The output of this stage is a new population (GASH's population) which is the input for the second layer (FC). This latter is based on an unsupervised fuzzy clustering algorithm, which performs the partition of the individuals into a set of $C$ clusters so that each of them corresponds to a niche. The number of clusters ($C$), as well as the characteristics of each cluster are automatically calculated by this layer without requiring any prior knowledge on the population to classify.

The final layer (SS) implements the principle of spatial separation to generate sub-populations from the resulting cluster characteristics (center, radius). This procedure yields a new and more appropriate population (SS's population) for the next cycle.

To validate the clustering results, the partition entropy ($h$) is used as a validity criterion, and the whole procedure is iterated until $h$ reaches a prefixed minimum value $h_{\text{min}}(<10^{-3})$. Hence, individuals are subject to a cyclical processing throughout the three layers of the system.

The system is based on the following points:

1. each cluster represents a niche;
2. cluster centers represent the expected optima;
3. as the fuzzy clustering algorithm (FCA) uses a self-learning procedure, the number of clusters ($C$) is not pre-specified, but computed by the algorithm itself;
4. the sub-populations are generated using the characteristics (center and radius) of each cluster; and
5. the evolution of each sub-population is handled by a GASH, in order to identify a non-detected niche in the previous cycle.

The implementation of the different layers is described in the following sections.

4.1. The GASH layer

It is well known that the choice of a selection procedure in a GA with sharing may influence the stability of the algorithm (Säreni and Krähenbühl, 1998). For example, when tournament selection is used, special provisions must be made to promote stability (Oei et al., 1991). In this study, the stochastic remainder selection (SRS) and stochastic universal selection (SUS) are used. These techniques present the advantage of reducing bias in the selection process with minimum spread (Bäck, 1996).

The SRS with replacement allocates samples according to the integer part of the expected value of occurrences of each chromosome in a new population and where the chromosomes compete according to the fractional part for the remaining places in the population. The SUS is an improved version, which uses a single wheel spin. This wheel, which is constructed in the standard way (as in the roulette wheel selection), is spun with a number of equally spaced markers equal to the population size as opposed to a single one (Michalewicz, 1996).

The similarity metric used in all examples is based on the phenotype. The other GA parameters are:

- number of generations $N_g = 30$,
- population size $= 150$,
- 2-pt crossover with rate $= 0.9$,
- mutation rate $= 0.005$.

The population $P(t)$ is initialized once only in the whole search process at the first cycle. In the
intermediate cycles, the population will contain the sub-populations built by layers 2 and 3.

4.2. The fuzzy clustering layer

The algorithm, which uses a measure of similarity between individuals in order to separate them into different clusters, comprises two main phases. The first one is an unsupervised learning procedure (Bouroumi et al., 1996), which yields an initial fuzzy C partition of the individuals issued from the GASH layer by exploring them sequentially. The procedure starts by generating the first class around the first individual encountered. Then a new cluster is automatically created each time the current object presents a small similarity, i.e., less than a specified threshold $S_{\text{min}}$, to all the already existing cluster centers. To measure the similarity between two vectors $x_i$ and $x_j$ of $R^p$, the following expression is proposed:

$$S(i, j) = 1 - \frac{d(x_i, x_j)}{\sqrt{p}}, \quad (10)$$

where $d(x_i, x_j)$ is a distance measure, calculated on the basis of the normalized values of $x_i$ and $x_j$. To normalize the $k$th component of each vector ($1 \leq k \leq p$) the following relation is used:

$$x_i^k \rightarrow \frac{x_i^k - \min(k)}{\max(k) - \min(k)}, \quad (11)$$

$\min(k)$ and $\max(k)$ are respectively the minimum and maximum values of the parameter $k$ along the set $X$ of $n$ samples, i.e.,

$$\min(k) = \min_{1 \leq i \leq n} \{x_i^k\}, \quad (12a)$$

$$\max(k) = \max_{1 \leq i \leq n} \{x_i^k\}. \quad (12b)$$

In Eq. (10) when $x_j$ is replaced by a cluster prototype $\mu_j$, the relation is also interpreted as the membership degree of $x_i$ to the $j$th cluster:

$$\mu_{ij} = 1 - \frac{d(x_i, \mu_j)}{\sqrt{p}}. \quad (13)$$

At each iteration of the learning process, the relation 13 is used for calculating the membership degree of the current pattern to all $C$ existing clusters ($C$ is variable). The condition of creating a new cluster is

$$\max_{i \neq j \leq C} (\mu_{ij}) < S_{\text{min}}. \quad (14)$$

When Eq. (14) does not occur, the centers of all existing clusters are updated according to:

$$\mu_{ij} = \frac{\sum_{k=1}^{i} \mu_{ik} x_k}{\sum_{k=1}^{i} \mu_{kj}}, \quad 1 \leq j \leq C, \quad (15)$$

where $i$ is the current element index.

The second phase of this layer is an optimization procedure that ameliorates the learned partition, generated during the previous phase. It uses the principle of the well-known fuzzy C-means algorithm described in Section 3.2.

This clustering algorithm is, of course, sensitive to the choice of the similarity threshold value. This means that different choices of this parameter may lead to different results. To solve this problem, the partition entropy defined by Eq. (4) is used as a validity criterion. According to this criterion, the best solution $U^*$ is the one that minimizes $h$. Once this solution is achieved, a defuzzification procedure is performed in order to affect definitely each individual to its natural group, i.e., the class for which it presents the maximum membership degree. This results in a final hard C partition with $C$ cluster centers $\mu_j$ ($1 \leq i \leq C$), which represent the expected optima. For each cycle, the parameters calculated for each detected cluster $C_i$ ($1 \leq i \leq C$), are:

- cardinal ($N$), i.e., the number of individuals;
- center ($\mu_j$): the mean vector of the cluster elements;
- radius ($r_i$) of the cluster;
- simax: the maximum of similarity between individuals of cluster $C_i$; and
- simin: the minimum of similarity between individuals of cluster $C_i$.

These parameters are of great use to follow the evolution process of niches. For example, $\mu_j$ and $r_i$ are used to define the niche $i$. Internal similarities indicate how the individuals occupy the niches.
throughout the cyclical process. As the individuals of the same cluster tend to concentrate around the center of that cluster, they become more identical, the internal similarities increase and the radius decreases.

4.3. Spatial separation layer

The main goal of this layer is to induce local geography in the population and force local competition within this structure. It involves the formation of sub-populations (demes) using results of the clustering procedure. At each cycle, a sub-population is formed using the center and the radius of each cluster. We note that a cluster radius is measured according to the most distant element of the center. Furthermore, the demes communicate with other through migrations of individuals (Herrera et al., 1997). A migration operator exchanges individuals between sub-populations, allowing new diversity to be injected into converging sub-populations. This operator is performed, between neighboring niches, in the first cycles, i.e., when the niches are not well split.

4.4. Time complexity

It is well known that the time complexity of the sharing method is $O(n^2)$. The time complexity added by the fuzzy clustering layer corresponds to the computation of the $c$ cluster centers and the distance between each of the $n$ data vectors and each center. Both steps can be done in $O(np)$ time. Furthermore, the determination of each new partition during the iterating process requires the calculation of the membership degree of each data vector to each class. This step is done in $O(np)$ time. Since the procedure is iterated until the algorithm has converged, the time complexity of the fuzzy clustering algorithm is in order of $O(npct)$ where $t$ is the number of iterations.

However, for many real-world applications, the dominant cost is that of calculating the fitness value for each population member for each generation (Mahfoud, 1995; Horn, 1997) and the time complexity $O(n^2)$ or $O(npct)$ can be small compared with the cost of evaluation of objective function.

4.5. Performance criteria

The last step of each cycle is concerned with the ability of the algorithm to identify the optima. For this, three different criteria are used.

4.5.1. Maximum peak ratio (MPR)

The maximum peak ratio statistic is used to indicate both the quality and the number of the optima identified. It is the sum of the fitness of the identified optimum divided by the sum of the fitness of the actual optimum in the search space:

$$\text{MPR} = \frac{\sum_{i=1}^{C} F_i}{\sum_{k=1}^{q} f_k},$$

where $F_i$ is the fitness of identified optimum $i$ and $f_k$ is the fitness of the actual optimum $k$. $C$ represents the number of identified clusters, which contain the identified optima, and $q$ is the number of real optima. An optimum is considered to be detected if its fitness value is at least 80% of the actual optimum (Miller and Shaw, 1995; Säreni and Krähenbühl, 1998).

4.5.2. Effective number of peaks maintained (EPM)

The EPM represents the ability of a niching GA (NGA) to locate and maintain individuals at the fitness peaks for extended periods of time. Most researchers simply track the number of peaks maintained by an NGA as a function of the number of evaluations performed (Deb and Goldberg, 1989; Säreni and Krähenbühl, 1998).

4.5.3. Number of fitness function evaluations

The effective function evaluations (NFE) measure is the number of function evaluations required for population convergence. Several methods to determine NGAs convergence have been introduced. The simplest one terminates an NGA when a fixed number of function evaluations have been performed. Beasley et al. (1993a) and Mahfoud (1995) use a halting window to determine NGA convergence. This mechanism halts an NGA when the average raw fitness stagnates. In this study, the algorithm converges when the entropy measure is less than $10^{-3}$. 
## 5. Experimental study

To illustrate the performance of our method, we present in this section the results obtained for a set of four known test functions. The experiments were performed using the multimodal functions F1, F2, F3 and F4 used by Beasley et al. (1993a), Mahfoud (1995) and Sareni and Krähenbühl (1998). These functions have several characteristics, which make them ideal for testing the ability of a GA to identify optima in a multimodal domain. As the location of each optimum is known, it is easy to compare the ending population distribution with the ideal population distribution. Note that in the clustering process, each object is defined by its coordinate(s) and fitness. Due to the goal of this study and limited space, we present only the critical cycles of the process.

### 5.1. Numerical results and discussion

**Example 1**

The first function (Fig. 2) is defined by:

\[
F_1 = \sin(5\pi(x^{0.75} - 0.05)).
\]

Over the interval [0, 1], the function has five peaks at \(x = 0.08, 0.247, 0.451, 0.681\), and 0.934. All these maxima have the function value of 1.0.

Table 1 shows the evolution of the entropy criterion, for different values of \(C\) (number of niches detected), throughout the cycling process. For each cycle, the best partition is obtained for the minimum entropy (bold in Table 1). For example, in the second cycle, the minimum entropy (0.033) is obtained for a 6-partition \((C^*=6)\). We note also that for some cycles, some values of \(C\) are not detected.

<table>
<thead>
<tr>
<th>(C)</th>
<th>First cycle</th>
<th>Second cycle</th>
<th>Third cycle</th>
<th>Fourth cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.412</td>
<td>-</td>
<td>0.239</td>
<td>0.234</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>0.310</td>
<td>0.353</td>
<td>0.346</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>0.193</td>
<td>-</td>
<td>0.201</td>
</tr>
<tr>
<td>5</td>
<td>0.595</td>
<td>-</td>
<td>0.001</td>
<td>(10^{-6})</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>(0.033)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>0.643</td>
<td>0.237</td>
<td>0.056</td>
<td>-</td>
</tr>
</tbody>
</table>

In the second cycle, the clustering algorithm does not provide 2-partition and 5-partition.

**Example 2**

The second function is

\[
F_2 = \exp\left[-2 \log(2) \left(\frac{x - 0.08}{0.854}\right)^2\right] \sin(5\pi(x^{0.75} - 0.05)).
\]

It is an oscillating function with five peaks of
The peaks are located approximately at the values of \(x\): 0.08, 0.247, 0.451, 0.681 and 0.934. Rounded values of these maxima are respectively 1.0, 0.948, 0.770, 0.503 and 0.250.

Results obtained for this example are summarized in Table 3. The analysis of these results shows that in the three first iterations when the algorithm has not yet converged, the entropy is relatively high, e.g., in the second iteration the minimum of entropy \(h=0.0526\) is obtained for \(C^*=4\). The first cluster \((u_1=(0.131, 0.981))\) contains two optima corresponding respectively to the \(x\) values 0.080 and 0.247. This is confirmed in the next cycle where the two clusters \((u_1=(0.082, 0.999)\) and \(u_2=(0.242, 0.942)\) are split and five clusters identified. At the fourth cycle, the entropy is less than \(10^{-5}\). As indicated by MPR, the best distribution is then obtained in this cycle. Due to the decrease of different clusters radius and the increase of the internal similarities, we note that the individuals are progressively grouped around peaks. For example, in the first cycle we have \(S_{\text{min}}=0.854\) and radius = 0.168 for the cluster centered at \((0.077, 0.957)\). At the fourth cycle \(S_{\text{min}}=0.999, r=0.0001\) and the new center becomes \((0.080, 1.00)\). Theoretically, in the last cycle, when the algorithm has converged the individuals in the same niche are identical and internal similarities become close to 1.

As reported in Table 3, the last column shows also that at fourth cycle internal similarities of all clusters are very close to 1 regardless of their values in the first cycle. These values can be very different because initial population is randomly generated. This effect is confirmed in Fig. 3, which shows the placement of individuals in population on search space.

**Example 3**

Let us examine now the efficiency of the GASH-FC on a deceptive function. The deceptive function (Fig. 4) considered here is defined, in the interval \([0, 6]\). This function has one local maximum at \(x=3\), with a fitness of 0.6, and two global maxima located at \(x=0\) and \(x=6\) with fitness of 1.

Table 4 shows that the three clusters detected, in the first iteration, do not overlap. In fact, even the
optima are identified, the following cycles permit a fine tuning of this optima. This is confirmed by the MPR, which is equal to 0.895 in the first cycle and to 0.9992 in the fifth cycle.

**Example 4**

The function $F_4$ is the modified Himmelblau’s function, given below and shown in Fig. 5 (Beasley et al., 1993a):

$$F_4(x, y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2.$$  

This function has four maxima of equal height (200) at approximately $(3.58, -1.86), (3.0, 2.0), (-2.815, 3.125)$ and $(-3.78, -3.28)$. The variables $x, y$ are both defined in the range $-6 \leq x, y \leq +6$. The optima of $F_4$ are more difficult to identify and locating them all is not immediate. In fact, only two optima are found in the first three sequences.

Because of the overlapping clusters, the two other optima do not appear. Fig. 6 shows that the entropy ($h$) increases during the three first cycles. As the clusters are dynamically adjusted, $h$ is decreased at cycle 4 when all optima are located. Full convergence occurs by cycle 6 when all elements are distributed on all optima. The MPR (0.9998) indicates that the identified optima are very close to the actual optima. The two-dimensional representations (Fig. 7) illustrate the placement of individuals in the search space and thereby the formation of clusters during the cycling process.

Fig. 3. The distribution of population elements is shown for four cycles.
It should be noted that even though the final result is the same for different executions of GASH-FC (for all test functions), the intermediate results, i.e., results at the end of each cycle, may change considerably from one execution to another. This is due of course to the random nature of the initial population in a GA.

To summarize, we present in Fig. 8 the evolution of the number of peaks maintained as a function of NFE for the four examples studied.

5.2. Comparisons with other NGAs

To allow comparisons between our model and other NGAs, this section presents the results obtained for functions F1 and F2 by some typical NGAs including GASH-FC. Table 5 displays, for example, values of the three performance criteria using the GASH-FC and three other models described in Sareni and Krähenbühl (1998). The ‘sharing + sort’ implements the fitness sharing coupled with a matching sort algorithm.

The efficiency of the niching method is related to its capacity to find new niches without recalling the niches already identified as well as preserving the best individuals during the search process. We can see that the GASH-FC and clearing models are the best NGAs realizing this compromise.

Note that the NFE performed in our model is not fixed a priori but checked when the algorithm converges. In contrast, for the three other NGAs, this parameter is fixed by the number of generations that may be performed so that the algorithm converges.
6. Conclusion

The goal of the approach proposed in this paper is to overcome some of the problems associated with the sharing technique in multimodal domains. It presents the advantages of forming sub-populations, which can evolve in parallel and explore separate portions of the search space, without requiring prior
knowledge of the initial population. Furthermore, since the niche radii are dynamically adjusted, fine local tuning is improved allowing successive amelioration of the solutions during the cycling process. That is, the proposed approach is able to find new niches without recalling the optima already located. Although this version of the model could be characterized more accurately as quasi-coevolutionary, it would be particularly suited to a parallel implementation in which each sub-population is evolved on a separate processor.

**Acknowledgements**

The authors are grateful to the anonymous referees for their helpful advice in improving the manuscript.
Table 5
Performance criteria obtained by different models for F1 and F2 functions

<table>
<thead>
<tr>
<th>Niching models</th>
<th>Maximum peaks maintained</th>
<th>Maximum peaks ratio</th>
<th>NFE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>F2</td>
<td>F1</td>
</tr>
<tr>
<td>GASH-FC</td>
<td>5</td>
<td>5</td>
<td>1.0</td>
</tr>
<tr>
<td>Sharing + sort</td>
<td>5</td>
<td>4.8</td>
<td>0.99</td>
</tr>
<tr>
<td>Clearing procedure</td>
<td>5</td>
<td>5</td>
<td>1.0</td>
</tr>
<tr>
<td>Deterministic crowding</td>
<td>5</td>
<td>4</td>
<td>1.0</td>
</tr>
</tbody>
</table>

References