A Genetic Algorithm with Sharing Scheme using Fuzzy Adaptive Clustering in Multimodal Function Optimization

Wagner F. Sacco*, Marcelo Dornelles Machado*, Roberto Schirru*
*Universidade Federal do Rio de Janeiro - PEN/LMP - COPPE/UFRJ
Caixa Postal 68 513 - 21945-970 - Rio de Janeiro – RJ - Brasil
{sacco, dornelas, schirru}@lmp.ufrj.br

Abstract. Genetic Algorithms (GAs) are systems based upon principles from biological genetics that have been used in function optimization. However, traditional GAs have shown to be inadequate in some cases, specially multimodal functions. Niching Methods allow genetic algorithms to maintain a population of diverse individuals. GAs that incorporate these methods are capable of locating multiple, optimal solutions within a single population. This paper describes a niching technique for GAs based on a fuzzy clustering method. The obtained results are presented using four different multimodal functions. The results show that the new method is quite promising, having potential to be applied in real world multiple solution problems.

1. Introduction

Genetic Algorithms [Holland 1975] (GAs) are systems based upon principles from biological genetics that have been used to solve complex problems with few a priori information, as function optimization or combinatorial optimization problems.

However, traditional GAs have shown to be inadequate to complex multimodal function optimization, as they rapidly push an artificial population toward convergence. That is, all individuals in the population soon become nearly identical. Niching Methods [Mahfoud 1995] allow genetic algorithms to maintain a population of diverse individuals, locating multiple, optimal solutions within a single population.

This paper describes a niching technique for genetic algorithms based on a fuzzy clustering method known as FCM (Fuzzy Clustering Means) [Bezdek 1981]. The GENESIS genetic algorithm tool is used. First, niching methods are introduced, giving emphasis to sharing, which is the universally adopted technique. Then, a new method, sharing with FCM, is introduced. The obtained results are presented using four different multimodal functions (the M1 and M2 benchmark functions [Goldberg and Richardson 1987] and two proposed variations of these functions) and compared with the standard sharing method results. Finally, the concluding remarks are made.

2. The Niching Methods

Genetic Algorithms have proven to be efficient in a great variety of domains, as the population of candidate solutions converge to a single global optimum. However, in multimodal domains, it may be interesting not only to find the global optimum, specially
in functions with equal optima. The convergence to a single optimum in this case is the result of the phenomenon known as genetic drift [Goldberg 1989].

Many populational diversity mechanisms have been proposed to force the GA to maintain a heterogeneous population throughout the evolutionary process, avoiding the convergence to a single peak. These mechanisms, called niching methods, allow the GA to identify, along with the global optimum, the local optima in a multimodal domain.

The analogy with nature is straightforward, as in an ecosystem there are different subsystems (niches) that contain many diverse species (subpopulations). The number of elements in a niche is determined by its resources and by the efficiency of each individual in taking profit of these resources.

Using this analogy, it is possible for the GA to maintain the populational diversity of its members in a multimodal domain. Each peak of the multimodal function can be seen as a niche that supports a number of individuals directly proportional to its “fertility”, which is measured by the fitness of this peak relatively to the fitnesses of the other peaks of the domain.

The difficulty in implementing niching methods lies on the fact that the peaks are obviously not known beforehand. This complicates the process of populating each niche correctly according to its fitness.

Various niching formation methods have been proposed, being generally based on sharing, which is the most disseminated paradigm and serves as reference when considering these methods.

Sharing. Fitness sharing was introduced by Goldberg and Richardson (1987), consisting of the reduction of the fitness of an individual proportionally to the number of nearby individuals. As selection occurs after the fitness correction, sharing does affect this mechanism.

The shared fitness \( f_i' \) of the \( i \)-th individual is given by

\[
 f_i' = \frac{f_i}{m_i'} \tag{1}
\]

where

\( f_i \rightarrow \) original fitness of the \( i \)-th individual.

\[
 m_i' = \sum_{j=1}^{N} sh(d_{ij}) \tag{2}
\]

is the niche count, that takes into account the whole population in relation to the \( i \)-th individual, derating the fitness of this individual according to the nearness of the others. The function that quantifies this proximity is the following:

\[
 sh(d) = \begin{cases} 
 1 - \left( \frac{d}{\sigma_{\text{share}}} \right)^\alpha, & \text{if } d < \sigma_{\text{share}} \\
 0, & \text{otherwise} 
\end{cases} \tag{3}
\]
where $\sigma_{\text{share}}$ and $\alpha$ are constants.

From eq. (2), it can be noticed that it takes $O(N^2)$ calculations per generation to find the derated fitnesses of the $N$ individuals of the population. Consequently, this method is computationally slow.

Sharing with cluster analysis. This method was created to overcome the slowness of standard sharing and not to depend on the troublesome estimation of $\sigma_{\text{share}}$ [Yin and Germay 1993]. It is based on the KMEAN adaptive clustering method [Anderberg 1975]. A cluster corresponds to a niche and the fitness of an individual is derated proportionally to the size of the cluster where this individual lies. It is necessary to specify parameters $d_{\text{min}}$, $d_{\text{max}}$ and the initial number of clusters $k_{\text{ini}}$. Clusters must be separated at least by $d_{\text{min}}$ and new clusters are created if there are individuals farther than $d_{\text{max}}$ from the existing clusters.

3. Sharing with FCM

As seen at the previous section, to group the GA elements in clusters that gather individuals with a high degree of “natural association” is a decisive step toward the formation of niches. Each individual is assigned exclusively to a single cluster by the KMEAN adaptive algorithm. But a question arises here: can we assure that an individual definitely belongs to the cluster to which it was assigned?

The classification scheme of the KMEAN algorithm is deterministic, being far from ideal if we have functions with unknown behavior or even with many adjacent extremes. In other words, the assignment of an individual to a cluster in real world functions has a high degree of uncertainty. The use of fuzzy logic [Zadeh 1965] to group the GA individuals in clusters appeared, thus, as a natural step to be taken. So, we applied to the GA the fuzzy class separation algorithm known as FCM (Fuzzy Clustering Means) [Bezdek 1981].

FCM became a popular algorithm exactly because it did not require, at each iteration, the total allocation of an individual to a certain cluster or class (using fuzzy logic terminology). This algorithm borrowed from fuzzy logic the concept of pertinence, that denotes the degree of association of an individual to a given class. In the case of KMEAN, pertinence $\mu_k$ of an individual $k$ to a class $i$ among $C$ classes is given by

\[
\mu_k = \begin{cases} 
1, & \text{if } d_{il} < d_{ik}; 1 \leq l \leq c, l \neq i \\
0, & \text{otherwise}
\end{cases}
\]

where $d_{ik}$ is the distance of individual $k$ to the centroid of class $i$. This means that, in KMEAN, the fact that an individual belongs to a given class or cluster excludes its pertinence to all the other classes.
In a fuzzy classifier, pertinence $\mu_{ik}$ must satisfy the following conditions:

$$\mu_{ik} \in [0,1] \text{ for every } i \text{ and } k,$$  \hspace{1cm} (5)

$$0 < \sum_{k=1}^{N} \mu_{ik} < N \text{ for every } i,$$  \hspace{1cm} (6)

$$\sum_{i=1}^{C} \mu_{ik} = 1 \text{ for every } k.$$  \hspace{1cm} (7)

Equation (7) is called the probabilistic restriction, since it states that the sum of the pertinences of an individual to all the existing classes must totalize 1 (or 100%).

Exactly like in KMEAN [Yin and Germay 1993], in FCM the processes of representation and assignment are alternated iteratively.

**a – Process of representation:** a class $i$ is represented by its centroid $c_i$ given by [Krishnapuram and Keller 1993]:

$$c_i = \frac{\sum_{k=1}^{N} \mu_i^m \cdot x_k}{\sum_{k=1}^{N} \mu_i^m}, \ i = 1,...,C$$  \hspace{1cm} (8)

with $x_k = (x_{k1}, x_{k2}, ..., x_{kj})$.

In the above equation $j$ is the number of variables, $x_{kj}$ is the value of the $j$-th variable of the $k$-th individual, $x_k$ is the $k$-th individual and $m$ is the nebulosity degree, that can be varied between 1 and $\infty$. Hall et al (1999) claim that the optimal value of the latter variable is $m=2$. Notice that in KMEAN there is no dependency on $m$, as $\mu_{ik}$ receives either 1 or 0.

**b – Process of assignment:** each individual belongs to the class with the greatest proximity to it. We decided to express this maximum proximity by the greatest value of pertinence. In other words, an individual $k$ belongs to the class $i$ with the greatest $\mu_{ik}$.

The FCM algorithm implicitly minimizes the total quadratic error of the individuals inside a class, which is given by:

$$J(L,U) = \sum_{j=1}^{C} \sum_{k=1}^{N} \mu_{ik}^m \cdot d_{ik}^2,$$  \hspace{1cm} (9)

where $L = (c_1, c_2, ..., c_C)$ is the array of centroids of all classes and $U$ is the $C \times N$ matrix that contains all pertinences $\mu_{ik}$. This matrix is known as the $C$-partition fuzzy matrix, where $C$ is the number of classes fixed a priori.
Deriving (9) in relation to $\mu_{ik}$ and making it equal to zero, respecting constraints (5), (6) and (7), the following equation is achieved:

$$
\mu_{ik} = \frac{1}{\sum_{l=1}^{C} \left( \frac{d_{lk}}{m} \right)}
$$

(10)

This equation quantifies the pertinence of the $k$-th individual to the $i$-th class. Notice that it takes into account not only the distance of the $k$-th individual to the $i$-th class, but also the distance of this individual to all the other classes.

The FCM algorithm is, basically, the following:

1- Fix a number of classes $C$; fix $m$, $1 < m < \infty$ (generally $m=2$);
2- Set the iteration counter value to $l = 1$;
3- Initialize the $C$-partition fuzzy matrix $U^{(0)}$;
4- Repeat
   Evaluate $c_i$ using equation (8);
   Update the elements of $U^{(l)}$ using equation (10);
   Increment $l$;
Until ($\| U^{(l-1)} - U^{(l)} \| < \varepsilon$).

Exactly like sharing with cluster analysis, sharing with FCM is applied to the GA population prior to the selection mechanism. The FCM algorithm groups the individuals in clusters or classes, assigning to each individual the class with the greatest value of pertinence.

The next step is fitness derating, where the individuals that occupy crowded niches are punished, favoring the emigration of their descendants. As we employ the sharing technique, a fitness derating term $m_i'$ must be found. The fitness derating of sharing with FCM, below, was based on sharing with clustering [Yin and Germay 1993]:

$$
\bar{f}_i' = \frac{f_i}{m_i'},
$$

(11)

with

$$
m_i' = n_c - n_c \left( \frac{d_{ic}}{2 \cdot d_{\text{min pert}}} \right)^\alpha \ x_i \in C_C,
$$

(12)
where $\alpha$ is a constant that usually receives 1, $n_c$ is the number of individuals in the niche or cluster to which the $i$-th individual belongs, $d_i$ is the distance of this individual to the centroid of its cluster and $d_{\text{minpert}}$ is the distance between the centroid of the cluster to which the $i$-th individual belongs and the individual of this cluster with the smallest pertinence value. Variable $d_{\text{minpert}}$ defines the cluster radius, as the individual with the smallest pertinence is the one farthest from the centroid.

A great asset of sharing with FCM lies on the fact that this method does not rely on pre-fixed constants, as sharing with clustering does ($d_{\text{max}}$ and $d_{\text{min}}$). These parameters are extremely sensitive to the function to be optimized [Mahfoud 1995]. It is necessary, though, to prefix the number of classes or clusters. This number, in the case of functions with unknown search space, would be simply the desired number of solutions (the top ten values, for instance).

4. The Test Functions

We employed in our initial tests the most widely used test functions when dealing with niching methods [Mahfoud 1995]: M1 and M2, respectively shown in Figure 1, below.

As standard sharing works with the niching radius $\sigma_{\text{share}}$ estimated beforehand, this method would have serious difficulties when dealing with niches of variable sizes. This would not be the case of sharing with FCM, where niches are defined adaptively, using as only information their desired number. Furthermore, if a search space has peaks heterogeneously disposed, standard sharing would not work out, as a large value of $\sigma_{\text{share}}$ would encompass many peaks in a single niche or, conversely, a small value of $\sigma_{\text{share}}$ would create many niches in a single peak. Sharing with FCM would be capable of creating the niches correctly.

To prove our remarks, we modified M1 and M2, creating two test functions that have both niches of variable sizes and peaks heterogeneously disposed. These functions, called Modified M1 and Modified M2, with the following equations.
Modified $M_1 = \begin{cases} \sin(5\pi x); & 0 \leq x < 0.4 \text{ and } 0.6 \leq x \leq 1.0 \\ \sin(20\pi x); & 0.4 \leq x < 0.6 \end{cases}$ \hspace{1cm} (13)

Modified $M_2 = \begin{cases} e^{-2\ln2\left(\frac{x-0.1}{0.8}\right)^2} \sin(5\pi x); & 0 \leq x < 0.4 \text{ and } 0.6 \leq x \leq 1.0 \\ e^{-2\ln2\left(\frac{x-0.1}{0.8}\right)^2} \sin(20\pi x); & 0.4 \leq x < 0.6 \end{cases}$ \hspace{1cm} (14)

are displayed in Figure 2.

![Figure 2. Test functions Modified M1 and Modified M2.](image)

5. Testing Results

Evaluation Criterion. The main objective of niching methods is to populate all the peaks of a function, avoiding convergence to a single extreme. Consequently, the main evaluation criterion must be the population distribution, which, in our tests, we accounted by percent in each peak.

The testing GA. To test sharing with FCM versus standard sharing, we employ the GENESIS [Greffenhette 1990] GA package, to which we attached a module with these niching methods. We use binary coding, 32 bit strings and the distances between individuals are measured phenotypically (euclidian distance).

The GA parameters are the usual when testing niching methods [Goldberg and Richardson 1987]: Population Size = 100 individuals; Crossover Rate = 0.9; Mutation Rate = 0.0; Generation Gap = 1.0; Number of Generations = 100 and no elitism. To standard sharing, we apply the usual $\sigma_{\text{share}} = 0.1$ to $M_1$ and $M_2$ and to our new functions, following the same logic, we set $\sigma_{\text{share}} = 0.025$.

Sharing with FCM vs. Standard Sharing. We compare sharing with FCM with standard sharing, which is the most disseminated niching method.

Our results are averages of executions of each case using always the same random seed set.
Table 1, below, shows the results for test function M1. Sharing with FCM concentrates more individuals in the peaks than standard sharing, but the latter technique distributes the population slightly more uniformly.

<table>
<thead>
<tr>
<th>Peak</th>
<th>Sharing with FCM</th>
<th>Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>21.1</td>
<td>16.8</td>
</tr>
<tr>
<td>#2</td>
<td>18.3</td>
<td>17.7</td>
</tr>
<tr>
<td>#3</td>
<td>17.5</td>
<td>16.7</td>
</tr>
<tr>
<td>#4</td>
<td>18.9</td>
<td>16.6</td>
</tr>
<tr>
<td>#5</td>
<td>17.6</td>
<td>18.3</td>
</tr>
<tr>
<td>Total</td>
<td>93.4</td>
<td>86.1</td>
</tr>
</tbody>
</table>

Table 1. Percent of individuals per peak in M1.

In test function M2, as Table 2 illustrates, sharing with FCM inserts more individuals in the peaks than traditional sharing. Niching methods are generally not capable of populating the fifth peak of M2 [Watson 1999], but the new method manages to do it.

<table>
<thead>
<tr>
<th>Peak</th>
<th>Sharing with FCM</th>
<th>Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>29.0</td>
<td>27.5</td>
</tr>
<tr>
<td>#2</td>
<td>25.8</td>
<td>26.5</td>
</tr>
<tr>
<td>#3</td>
<td>19.2</td>
<td>18.0</td>
</tr>
<tr>
<td>#4</td>
<td>11.0</td>
<td>9.5</td>
</tr>
<tr>
<td>#5</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Total</td>
<td>87.0</td>
<td>81.5</td>
</tr>
</tbody>
</table>

Table 2. Percent of individuals per peak in M2.

In Modified M1, as shown in Table 3, sharing has more individuals in the peaks than sharing with FCM, but without any symmetry and even without populating two peaks. The new method, on the other hand, populates all peaks with a good symmetry. It is worth mentioning that the small central peaks have less individuals than the other peaks. This is because these small peaks “compete” with each other in a small fraction of the search space to receive individuals.

<table>
<thead>
<tr>
<th>Peak</th>
<th>Sharing with FCM</th>
<th>Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>16.1</td>
<td>31.8</td>
</tr>
<tr>
<td>#2</td>
<td>15.9</td>
<td>43.5</td>
</tr>
<tr>
<td>#3</td>
<td>8.3</td>
<td>5.5</td>
</tr>
<tr>
<td>#4</td>
<td>2.9</td>
<td>3.7</td>
</tr>
<tr>
<td>#5</td>
<td>6.6</td>
<td>0.0</td>
</tr>
<tr>
<td>#6</td>
<td>8.4</td>
<td>0.0</td>
</tr>
<tr>
<td>#7</td>
<td>15.1</td>
<td>2.2</td>
</tr>
<tr>
<td>#8</td>
<td>17.2</td>
<td>7.8</td>
</tr>
<tr>
<td>Total</td>
<td>90.5</td>
<td>94.5</td>
</tr>
</tbody>
</table>

Table 3. Percent of individuals per peak in Modified M1.
Table 4 displays the results for Modified M2. Sharing, again, inserts more individuals in the peaks, but only in the two tallest peaks. Sharing with FCM, by its turn, distributes the population very well, populating even in the smallest peak.

<table>
<thead>
<tr>
<th>Peak</th>
<th>Sharing with FCM</th>
<th>Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>22.2</td>
<td>64.3</td>
</tr>
<tr>
<td>#2</td>
<td>21.9</td>
<td>31.0</td>
</tr>
<tr>
<td>#3</td>
<td>11.8</td>
<td>0.0</td>
</tr>
<tr>
<td>#4</td>
<td>12.0</td>
<td>0.0</td>
</tr>
<tr>
<td>#5</td>
<td>2.8</td>
<td>0.0</td>
</tr>
<tr>
<td>#6</td>
<td>7.7</td>
<td>0.0</td>
</tr>
<tr>
<td>#7</td>
<td>8.9</td>
<td>0.0</td>
</tr>
<tr>
<td>#8</td>
<td>4.3</td>
<td>0.0</td>
</tr>
<tr>
<td>Total</td>
<td>91.6</td>
<td>95.3</td>
</tr>
</tbody>
</table>

Table 4. Percent of individuals per peak in Modified M2.

6. Conclusion
In this paper we introduce a new niching method to find the optima (global and local) of multimodal functions.

Our tests show that sharing with FCM is a very promising method, as results are extremely satisfactory and it does not require a previous knowledge of the search space, as standard sharing and sharing with clustering do (estimation of $\sigma_{share}$ in the former and $d_{min}$, $d_{max}$ in the latter) [Yin and Germay 1993]. The only input is the number of desired classes. As we intend to apply GA with niching to real world problems, this feature is extremely relevant.

Acknowledgement
Marcelo Dornellas Machado is supported by FAPERJ (Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro).

References


