A Clearing Procedure as a Niching Method for Genetic Algorithms

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 clearing procedure is a niching method inspired by the principle stated by J.H. Holland in 1975: the sharing of limited resources within subpopulations of individuals characterized by some similarities. But, instead of evenly sharing the available resources among the individuals of a subpopulation, the clearing procedure supplies these resources only to the best individuals of each subpopulation. The clearing is naturally adapted to elitist strategies. This can significantly improve the performance of genetic algorithms applied to multimodal optimization. Moreover the clearing procedure allows the GA to efficiently reduce the genetic drift when used with an appropriate selection operator. Some experimental results are presented for a massively multimodal deceptive function optimization.**

I. Genetic Algorithms and Multimodal Optimization

A simple genetic algorithm [1] (SGA) is suitable for searching the optimum of unimodal functions in a bounded search space. However, both experiments and analysis show that the SGA cannot 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 in such a way that each highest maximum of the multimodal function can attract one of them. These mechanisms are referred to as "niching methods" [2].

The clearing procedure described in this paper derives from the niching principle stated by J.H. Holland in 1975 [3]. A niche is characterized by a limited amount of renewal resources available for individuals which presents similarities. Each individual in a niche can consume a fraction of the available resources: the greater the subpopulation size of the niche, the smaller the fraction. This leads towards a steady state in which the subpopulation sizes are proportional to the amount of the corresponding available resources. D.E. Goldberg and J. Richardson [4] presented an implementation of this concept known as the "sharing method".

This paper first presents the clearing procedure and subsequently an elitist variant. This technique is then compared with the sharing method from the subpopulation size point of view. The behavior of the clearing procedure facing the genetic drift with the operators "Roulette Wheel Selection" (RWS) or "Stochastic Universal Selection" [5] (SUS), is also discussed. Finally, the test of the clearing procedure on a massively multimodal deceptive function is presented.

II. Clearing the Search Space

A. Principles

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

Each subpopulation contains a dominant individual: the one that has the best fitness. If an individual belongs to a given subpopulation, then its dissimilarity with the dominant is less than a given threshold σ: the *clearing radius*. 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 rather than sharing resources with the other individuals of the same niche as is done in the sharing method.

With such a mechanism, the niche of an individual is not generally known. In effect, it can be dominated by several winners. On the other hand, for a given population, the set of the winners is unique. This proposition is proved by induction: the individual that has the strongest fitness in the population is necessarily a winner. The winner and all the individuals that it dominates are then fictitiously removed from the population. We proceed in the same way with the new population which is then obtained. Thus, the list of all the winners is produced after a certain number of steps.

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 not generally unique. There is at least one reason to want capacities greater than 1: if the capacities are equal to the population size, the clearing effect vanishes and the search method becomes a standard GA. Thus, choosing capacities between one and the population size offers intermediate situations between the maximum clearing and a standard GA.

B. The clearing procedure

A simplified version of the clearing procedure is presented below in pseudo code. **P** and *n* are global variables. "*Sigma*" is the clearing radius, "*Kappa*" is the capacity of each niche. "*nbWinners*" indicates the number of winners of the subpopulation associated with the current niche. Population **P** can be considered as an array of *n* individuals.

```
function Clearing(Sigma, Kappa)

begin
  SortFitness(P)
  for i = 0 to n - 1
    if Fitness(P[i]) > 0
      nbWinners := 1
      for j = i + 1 to n - 1
        if Fitness(P[j]) > 0 and
            Distance(P[i], P[j]) < Sigma
          if nbWinners < Kappa
            nbWinners := nbWinners + 1
          else
            Fitness(P[j]) := 0.0
          endif
        endif
      endfor
    endif
  endfor

end
```

The simplified algorithm uses three functions:

- **SortFitness(P)** sorts population **P** according to the fitness of the individuals by decreasing values. The whole population is ranked for the sake of clarity in this version of the algorithm. A more optimized algorithm would sort only the dominant individuals.
- **Fitness(P[*i*])** returns a reference on the fitness of the *i*-th individual of population **P**.
- **Distance(P[*i*], P[*j*])** returns the distance between two individuals *i* and *j* of population **P**.

C. Complexity

In order to assign individuals to a given subpopulation, its dominant is compared to all the individuals that have not yet been assigned to a subpopulation. The dominant of subpopulation 1 is the first of the sorted population. It is compared to the *n*-1 other individuals to obtain subpopulation 1. Assume that this subpopulation contains I_1 individuals. Then, the dominant in subpopulation 2 is the first individual of the population not yet assigned. It is compared to the $n - 1 - I_1$ individuals that are not yet assigned. Thus, by induction reasoning, the dominant of subpopulation *i* has to be compared to the $n - 1 - \sum_{j=1}^{i-1} I_j$ individuals not yet assigned. Let *c* be the number of subpopulations, the total number of comparisons γ is:

$$\gamma = \left(\sum_{i=1}^{c} i \, I_i \right) - c$$

The maximal value of term $\sum_{i=1}^{c} i \, I_i$ is obtained when the subpopulation 1 contains $(n - c + 1)$ individuals, while each other subpopulation contains only one individual. Its minimal value is obtained when subpopulation *c* contains $(n - c + 1)$ individuals, and each other subpopulation contains only one individual. Consequently, γ belongs to the interval:

$$n + \frac{c\,(c - 3)}{2} \leq \gamma \leq c\, n - \frac{c\,(c + 1)}{2}$$

Moreover, the complexity of the sort is bounded by O(*c n*), since only *c* dominant individuals have to be ranked. Thus, the complexity of the clearing procedure is O(*c n*) in the worst case. When *n* increases, *c* tends towards a limit, which depends on the fitness landscape, σ and κ. These parameters should be chosen in such a way that the limit of *c* is of the same order as the desirable peak number.

D. Expected size of subpopulations under clearing

Let $w_X(g)$ be the number of winners of a subpopulation *x* at generation *g*. Let us assume that the fitness of the winners of a subpopulation are almost equal to a value f_X. This hypothesis is true for a capacity 1, or at the steady state when subpopulations are located on the peaks of the function. By applying a proportionate selection operator after a clearing, the expected size of each subpopulation x is

$$\mathrm{E}\left(n_x^{clearing}(g)\right) \approx n \frac{w_x(g)\, f_x}{\sum_{y=1}^{c} w_y(g)\, f_y} \tag{1}$$

where E(X) represents the expected value of random variable X. The experiments and the analysis [6] show that the niches associated with the best fitnesses tend to be saturated during the evolution.

An interesting particular case arises when all the niches are saturated and have an equal capacity κ. This is always the case when $\kappa = 1$. Expression (1) becomes

$$\mathrm{E}\left(n_x^{clearing}(g)\right) \approx n \frac{f_x}{\sum_{y=1}^{c} f_y} \tag{2}$$

In this case, the expected size of a subpopulation depends neither on generation g, nor on the niche capacity. On the other hand, the expected number of offspring E($n_{(i)}{}^{clearing}$) of individual i belonging to subpopulation x is inversely proportional to capacity κ:

$$\mathrm{E}\left(n_{(i)}^{clearing}\right) \approx \frac{\mathrm{E}\left(n_x^{clearing}\right)}{\kappa} \tag{3}$$

According to the selection operator used, this number has an effect on the genetic drift [1] due to the noise of the selection operator. The classical RWS does not ensure that the actual number of offsprings is close to the expected value. It is possible that an individual that has a fitness greater than the average disappears unfortunately. On the other hand, the SUS operator [5] ensures that for an expected number of offspring n_S, the actual number is $\lfloor n_S \rfloor$ or $\lceil n_S \rceil$.

The niche capacity has no consequence on the noise of an RWS. However, the smaller the number E($n_{(i)}{}^{clearing}$), the greater the relative noise of an SUS. A subpopulation is surely preserved if E($n_{(i)}{}^{clearing}$) ≥ 1, but it could disappear if a large enough niche capacity is selected. Thus, a high capacity modifies the exploration of the search space by allowing a large part of the population to concentrate towards the highest maxima of the fitness function.

E. 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 an unimodal fitness function, but also 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 supplies these individuals naturally: 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.

III. Clearing vs. Sharing

The sharing method [4] is probably the best known niching method. With a proportionate selection operator, the expected number of offspring of an individual i is proportional to its *shared fitness* $\tilde{f}_{(i)}$ which depends on its fitness $f_{(i)}$ according to the following expression:

$$\tilde{f}_{(i)}(g) = \frac{f_{(i)}}{\sum_{j=1}^{n} sh\left(d_{ij}\right)} \tag{4}$$

with

$$sh(d) = \begin{cases} 1 - \left(\dfrac{d}{\sigma_{share}}\right)^{\alpha}, \text{ if } d < \sigma_{share} \\ 0 \qquad \text{otherwise} \end{cases} \tag{5}$$

sh : sharing function

d_{ij} : genotypic or phenotypic distance between individuals i and j

σ_{share} : threshold of dissimilarity

α : constant (typically set to 1)

Consider the hypothesis where all the individuals of a subpopulation x have a fitness almost equal to a value f_X, representative of the niche and are at a distance near zero from each other. This particular case typically occurs at the steady state when subpopulations are localized on the highest maxima of the function. Assume that individual i belongs to subpopulation x, a new expression of $\tilde{f}$ is obtained:

$$\tilde{f}_{(i)}(g) \approx \frac{f_{(i)}}{n_x(g)} \approx \frac{f_x}{n_x(g)} \tag{6}$$

where $n_x(g)$ is the number of individuals in the subpopulation x at generation g. The expected number of offspring of an individual i at generation g is :

$$\mathrm{E}\left(n_{(i)}^{sharing}(\ g)\right) = n\ \frac{\tilde{f}_{(i)}(g)}{\sum_{j=1}^{n} \tilde{f}_{(j)}(g)} \approx \frac{n}{n_x(g)}\ \frac{f_x}{\sum_{y=1}^{c} f_y} \tag{7}$$

The expected number of individuals $\mathrm{E}\ \left(n_x^{sharing}\right)$ that belong to subpopulation x is obtained by multiplying expression (7) by $n_x(g)$:

$$\mathrm{E}\ \left(n_x^{sharing}\ \right) \approx\ n\ \frac{f_x}{\sum_{y=1}^{c} f_y} \tag{8}$$

This number is independent of g when the steady state is reached. Thus, the sharing method tends to restore the equilibrium at every generation. Note that the expression of $\mathrm{E}\ \left(n_x^{sharing}\right)$ is identical to that of $\mathrm{E}\ \left(n_x^{clearing}\right)$ when the niches are saturated and have the same capacities.

This second expression of the expected number of offspring of individual i at generation g gives another point of comparison with the clearing:

$$\mathrm{E}\left(n_{(i)}^{sharing}(\ g)\right) \approx \frac{\mathrm{E}\ \left(n_x^{sharing}\ \right)}{n_x(g)} \tag{9}$$

Thus, with the sharing method, if a subpopulation contains more individuals than expected because of the selection noise at a given generation, each of them will have an expected number of offspring less than 1 in order to restore the equilibrium. When this happens with proportionate selection schemes such as RWS or SUS, an individual of such a subpopulation could not have offspring. If this is the case for all the individuals of a subpopulation, it disappears. This phenomenon can be countered with a large enough number of individuals for each desirable subpopulation in such a way that its survival probability is high. This implies an adequate dimensioning of the population [8].

On the other hand, the clearing associated with an SUS ensures that the number of offspring of the winners of a subpopulation is always greater or equal to one if its fitness is greater than the average after clearing. Consequently, this subpopulation survives with certainty. Thus, this technique gives a lower bound to the population size which is smaller than the one required by the sharing method. This has been confirmed by the experiments.

IV. Experiments

Several experiments have been performed in order to evaluate the robustness of the clearing. Both easy and difficult fitness functions have been considered. The experiments described here concern the most difficult function that has been examined: the **M7** massively multimodal deceptive function used in [8] and [2] to test different niching methods.

The purpose of these experiments was to compare the performance of several variants of genetic algorithms on a binary-coded massively multimodal deceptive problem. Five variants of GA's are considered: an SGA, elitist or not, the sharing method associated with an SGA, and the clearing method, elitist or not, associated with an SGA.

A. Description of the **M7** *function*

M7 is defined as follows:

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

where $\forall\ k,\ x_k\ \in\ \{0,\ 1\}$. Function $\mathbf{u}(x)$ is defined for the integer values 0 to 6 (figure 1). 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 $\mathbf{u}$ has been specifically built to be deceptive.

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.

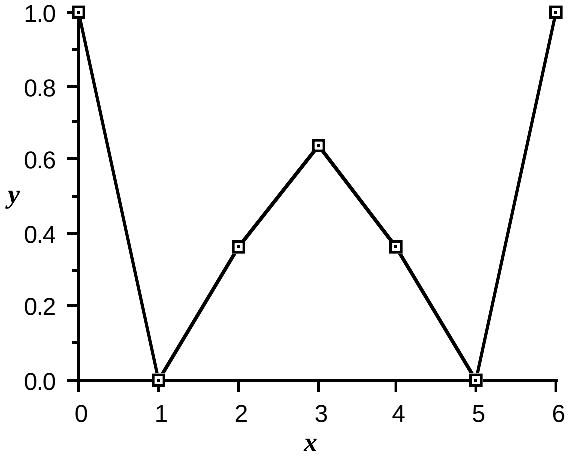

Figure 1: $y = \mathbf{u}(x)$

B. Experimental conditions

The 32 global maxima of this function are searched with a genetic algorithm working on 30-bit genotypes. The single-point crossover and standard binary mutation operators are used. The mated individuals are chosen at random, without constraint.

The distance between individuals is the Hamming distance between genotypes, normalized so that the biggest value in the search domain of the GA is equal to one. The clearing radius σ, as parameter σ_{share}, are chosen equal to 0.2 in the distance unit previously defined. This radius corresponds to the smallest distance that exists between two global maxima. This value was used in [8]. Parameter α of the sharing method is set to 1.

The population contains 600 individuals. The mutation rate is 0.002, the crossover rate is 1. These parameters have been chosen with the help of preliminary experiments in such a way that they minimize the duration of the computations.

C. Results

The performance is the mean number of peaks found by the GA at a given generation g for 100 different runs using the same parameters. A peak is found if at least one individual in the population corresponds exactly to a global maximum.

Figure 2 shows the performance vs. generation g for two clearing variants (elitist and non-elitist) with niches of capacity 1. These performances are compared to those yielded by the sharing method and two variants of simple genetic algorithms. The results are given for a stochastic universal selection (SUS).

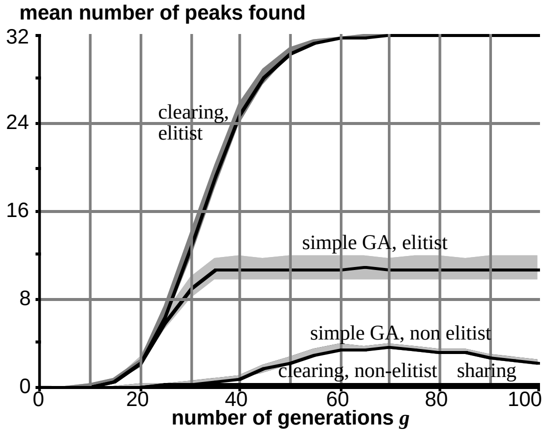

Figure 2: performance vs. the number of generations for searching the maxima of function **M7**. The greyed areas associated with each curve define the confidence intervals with an error probability of 1%.

The best results are obtained with elitist strategies. The clearing elitist strategy is the only which allows the GA to find all the 32 global maxima for less than 88 generations over 100 runs. On the other hand, the performance of the non-elitist clearing is near zero. This can be explained by the deceptiveness of function **M7**: the crossover destroys the individuals located on the peaks found with a high probability, while the search operators produce with low probability these individuals. The elitist techniques avoid this problem. However, the elitist simple GA does not give satisfaction according to frequent premature convergences.

The experiments show that the sharing method has a performance near zero for a population of 600 individuals. Concerning a hybrid implementation that associates a genetic algorithm with a hill-climbing, S.W. Mahfoud [2] wrote that the sharing method fails whatever the size of the population may be, in the limit of 1.5 million fitness evaluations. The best results were obtained with the “deterministic crowding” method, which required $101 \cdot 10^3$ evaluations on the average to find the 32 peaks. In comparison, an elitist clearing with $\kappa = 1$ associated with a SUS needs $22 \cdot 10^3$ evaluations on the average to solve the problem, with a minimum of $16 \cdot 10^3$ and a maximum of $32 \cdot 10^3$ evaluations measured over 100 runs.

The table below allows us to compare the number of peaks found by the clearing method at the hundredth generation with an SUS or an RWS and a niche capacity greater than 1, for 100 successive runs.

Nb. peaks g=100	SUS $\kappa = 1$	RWS $\kappa = 1$	SUS $\kappa = 8$	RWS $\kappa = 8$
elitist	32 ± 0	32 ± 0	31 ± 1	30 ± 1.4
non-elitist	0.1 ± 0.1	0.4 ± 0.2	3.8 ± 0.5	2.1 ± 0.5

The table shows that there is an advantage to accept several winners per niche for the non-elitist implementations. But this is a drawback for the elitist implementations. The explanations lie in the greater number of individuals located on the peaks found. These numbers obtained just after the fitness evaluation are given in the table below:

Nb. ind. on peaks	SUS $\kappa = 1$	RWS $\kappa = 1$	SUS $\kappa = 8$	RWS $\kappa = 8$
elitist	43	43	430	400
non-elitist	0.13	0.44	5.1	3.6

When the implementation is non-elitist, the crossover must destroy all the individuals of a subpopulation in order to suppress it. That is all the less likely as it has more individuals. The same phenomenon is at the origin of some peaks found by the non-elitist SGA.

By analyzing the developments of the tests for the clearing elitist implementations, one notes that the degradation is due to premature convergences towards 16 of the 32 peaks in 7 cases for the SUS, 13 cases for the RWS, over 100 runs. These premature convergences correspond to allele losses on one of the 6-bit blocks of the genotype. This happens because the large part of the population on the peaks already found no longer participates in the search process. This is due to the particular form of the **M7** function. The rest of the population is too small to regenerate the lost alleles likely.

The number of subpopulations at the hundredth generation is 364 ± 3 when the niche capacity is set to 1, and 34 ± 2 when this capacity is 8, run over 100 tests. In tests run with the latter parameter the number of subpopulations is almost equal to the number of global maxima.

V. Conclusion

The clearing procedure is an efficient niching method, which has the following properties:

- ◊ The complexity of the clearing procedure is lower than that of the sharing method. Clustering methods such as [9] could contribute to further speed up the algorithm.
- ◊ The clearing is directly compatible with elitist strategies.
- ◊ The genetic drift due to the selection noise is significantly reduced with the SUS operator, and the populations may be far smaller than those required by the sharing method.
- ◊ The behavior of the GA with the clearing procedure can be controlled between that of the maximum clearing and that without clearing by setting the niche capacity to an appropriate value.

This paper has presented the successful solving of a difficult multimodal problem with the clearing procedure compared to some other methods. Several other easy and difficult multimodal functions have also been tested. The clearing procedure has to date always outperformed the sharing method.

Acknowledgments

The author wishes to thank *K. English* for reading the manuscript. He is also grateful to the anonymous referees for their advice in improving it. This work was supported by *France Telecom*.

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] J.H. Holland, *Adaptation in Natural and Artificial Systems*, The MIT Press, 1992 (1st ed.: 1975)

[4] 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

[5] 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

[6] A. Petrowski, *Evolution of Subpopulations under the Clearing Procedure*, technical report, 1996

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

[8] 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

[9] W. Spears, “Simple Subpopulation Schemes”, *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, pp. 296-307, World Scientific, 1994