Alternative Strategies to Solution the Problem of Classification and to Design Genetic Operators

G. Winter, R. Berriel, I. Sánchez, and V. Gutiérrez*

CEANI, Universidad de Las Palmas de Gran Canaria
Edificio de Ingenieras, Campus Universitario de Tafira Baja
Tafira 35017, Las Palmas
Canarias España

1 Introduction

In this paper we deal with two different problems in relation with the research that we are development inside of CEANI in the frame of Genetic Algorithms (GA). In our research we, mainly, try to answer two questions, one of them is, given a problem of classification to find out an alternative solution to the classical methods of cluster analysis. Another one is, given a GA we are going to redesign some part in it, to optimise its efficiency.

2 New design in the classification problem

We focus our work on the environment of councils where we are trying to apply our own integrated, useful methodology to manage the intensive activities in information, which develop in the councils. For this reason we have designed as main aim classify the councils in accordance with a methodology. We point out that the importance of classifying the entities and obtaining different groups is based on the possibility that they are given, in accordance with the viewpoint of classification employed, to decide if any council needs a methodology to develop IS, because it could be that this application will be not necessary.

One of the most direct and used methods, at the moment is the cluster analysis, included in the statistic packages. *Had this the dangerous ability of create groups in the dates, even when don't exist natural clusters, for this reason they must be take into account with care*, Jain [3]. When we have obtain a partition is necessary validate it, generally, with apply the Variance Analysis, we must have a number of them $N \gg n$.

Due to the difficulties aforementioned, we propose a alternative methodology of work where we consider the GA. Our aim is to obtain the best classification of objects in interest groups, in front of the detention of geometric or hierarchic groups that can conduct us to a classification not relevant to the research that we are carrying out.

We identify the problem that we are going to work on find out the best partition of the set $\{O_1, O_2, \ldots, O_n\}$ where we represent each item O_i, for the vector of characteristics $(x_{1i}, x_{2i}, \ldots, x_{li})$. The criteria to decide the best partition is defined by the expert using a small set of simple rules $\{R_1, R_2, R_m\}$ that correctly represents a complex set of thoughts.

*Departamento de Estadística, Investigación Operativa y Computación. Universidad de La Laguna

To confront this problem we need to generate different situations that we can be in, this leads us to choose the GA in which the individuals will be the part of condition of the rule of decision. We establish, based on the experience that we have got, that must be in according with implementation, development, maintenance, and changes, so we obtain the next four groups:

Group 1 Grown from the councils where the implementation of IT and the development have been correct

Group 2 Grown from the councils, like the before case, the implementation has been correct, but the next development has not following the same line.

Group 3 Where la implementation has been not correct, but the development is being correct.

Group 4 Represent those councils where the implementation like the development are not correct.

We show in Table 1 the patterns that will permit us to define the fitness function, where high value of the characteristics are represented for 1, and low value for 0. Where the fitness function is:

$$
\begin{array}{lcl}
f : \mathcal{R}^n & \longrightarrow & \mathcal{R} \\
O = (x_1, \ldots, x_n) & \hookrightarrow & \max\left(f_1(O), \ldots, f_{K'}(O)\right)
\end{array}
$$

K' is the cardinal of the patterns set $\{P_1, P_2, \ldots, P_{K'}\}$ and $f_i(O) =$ *the number of concordances weighty between the pattern* P_i *and the individual* O.

Gr	1	2	2	3	3	4	4
X_1	1	0	1	1	1	0	1
X_2	1	0	0	1	1	0	0
X_3	1	0	1	0	1	0	0
X_4	1	1	1	0	0	0	0
X_5	0	1	1	1	0	0	1
X_6	0	1	0	1	0	0	1
X_7	1	0	0	1	1	0	0
X_8	0	1	1	0	0	0	0
X_9	1	1	0	1	0	0	0
X_{10}	1	1	0	1	0	0	0
X_{11}	1	1	0	1	0	0	0
X_{12}	1	1	1	1	0	1	0
X_{13}	1	1	0	0	0	0	0
X_{14}	1	1	0	1	0	0	1
X_{15}	1	0	1	0	1	1	1
X_{16}	1	1	1	1	0	1	0
X_{17}	1	0	0	1	1	0	0
X_{18}	1	1	0	1	0	0	1

Table 1: Definition of groups

Using an GA with proportional selection, cross in 2 points with probability of cross $p_c = 0.7$, probability of mutation $p_m = 3.81 * 10^{-2}$ and population of 5000 individuals.

From here we must determine in which one group are classify any council, we do this on the base of 18 questions that we provided in the questionnaire, answered by the technicals who work in the councils, that form the variables of classification.

We point out that the worse of them have a value of fitness function of 83125 above the maximum value that we can get of 100000. The validation has been made on the base of the proposal of classification and assignment of councils to different and interest groups in relation with the implementation and development of IT, carried out on 32 councils in the province of Las Palmas with a population less of 50000 inhabitants that we considered in this research. For this we developed a software that detect if any council, among the 32 that we tested is inside of a group of sorted entities, in according with our methodology.

3 Improving GAs efficiency introducing variance reduction

The robustness is one of the characteristics that has made the GA's an attractive tool for the optimisation problems. This robustness is due to the GA is an Black Box Optimisation (BBO) algorithm, ed. the objective function is often available like a black box: for a given x in the feasible domain, it returns the function value. No local or global information about the function is assumed. However, the GA's, like every BBO algorithm, need to explorer relations and classes between the search space members since another case we will not improve the random search, Kargupta and Goldberg [4]. A GAS implicitly divides the search domain into classes called schemas in the GA community, but not always in the best form.

We will employ the statistician knowledge obtained from previous populations to increase the GA efficiency without generality lost, by defining class and relations more appropriated. This knowledge will be employed adapting the exploration/exploitation balance so important in the genetic search. This paper is focused on improving GAs operator using knowledge generated from the GA evolution itself to increase convergence velocity. All these methods introduce variance reduction through changes on standard operators usually used in GAs. The ideas in this paper have been inspired from techniques of reduction variance (TVR) used to increase performance in Monte Carlo. The basic idea that will be developed consisting to use the information available to obtain good samples for the simulation (corresponding with fit individuals in GA's).

In this paper will be described along the following sections three strategies that have arisen like results from the aforementioned considerations: *discriminating acceptance of a crossover using a confidence interval* that will be denoted by ACI, Crossover Imposing Alleles (CIA Crossover) and a special selection operator for optimisation problems (SOOP). Finally we will present a section including different test functions that have been employed to evaluate our proposals.

4 Building Approximated Confidence Intervals (ACI)

We resort to consider samples of the population, whose characteristics allow us to induce properties that are interesting from the entire population. We use an approximated confidence interval to locate the subdomain of the search space where exits a high probability to find the best individuals of each generation. So we will employ the confidence interval to define an appropriated class to sesgate the search. In each generation of the genetic algorithm we have N individuals from which we want to extract the information that help us in the search of new *good* individuals. That is to say, we are interested in the random variable ξ= *"good individuals"*. From information contained in the sample about the random variable ξ we must extract the which one that allows us to construct a region where the parameter $\mu = E[\xi]$ is included with a probability of at least $1 - \alpha$. However, we really count on individuals in general to develop this task, ed., the sample arises from the random variable $\eta =$ *"individuals generated by the genetic algorithm"*. In general the average of that variable cannot be close to the optimum and loss of the effectiveness in the procedure can occur.

To save this difficulty, we propose to construct an *approximated confidence interval* for the variable ξ, taking best individuals of the population as sample and then we can guarantee that individuals are *good*. However, for sample size smaller than 30 we are making a mistake respect to suppose that the sampling average keeps a normal distribution. This has led us to consider a modification in the confidence interval definition, proposing the following approximated interval:

$$\left[\frac{1}{\beta N}\sum_{i=1}^{\beta N}\xi_i - k\frac{S_{n-1}}{\beta N}, \frac{1}{\beta N}\sum_{i=1}^{\beta N}\xi_i + k\frac{S_{n-1}}{\beta N}\right]$$

Here k is a constant and β represents the percentage of best individuals to be taken to build the interval, and they are defined in the applications. S_{n-1} is the *sample quasi-variance*. When the *class of good individuals*, corresponding to the confidence interval, has been constructed, it will be employed in the next generation according to the procedure that we have called *Discriminating Acceptance of Crossover*. The procedure is detailed in the next points:

1. It is supposed that we have already formed the $t - th$ generation of the genetic algorithm, then an approximated confidence interval is constructed.

2. We make the following generation applying selection operator.

3. Let be two individuals to be crossed. After crossover if at least one of the two individuals forming the offspring falls in the confidence region then the children resulting from the crossover will take part of the $t + 1$ generation, in another case the parents are considered to form part of the new generation.

Notice that the confidence interval is operating as a discriminant element that indicates when two children are accepted in the new generation. In other words, if the two children are bad they will not survive, since offspring will not improve the genetic information of its parents. They will do that, with efficiency only if at least one of them is enough and similar or better with high probability than unless a parent well adapted. Moreover when the restriction is imposed to only one descendant (not to both simultaneously), we are giving a degree of freedom diversity can operate, which is very important in the genetic algorithms. We formulate the discriminate acceptance defining the class $I = \left\{x \in S^l / x \in I\right\}$, where I is the approximated confidence interval. This class will be the class used as discriminant in the crossover operator.

Let be the function: $\delta : S^l \longrightarrow \{0, 1\}$ defined by

$$\delta(x) = \begin{cases} 1 & \text{if } x \in I \\ 0 & \text{if } x \notin I_i \end{cases}$$

All the steps of the procedure will be:

1. Select two individuals, the parents: x, y.

2. Make the crossover $x \otimes y$ and to decide on:
$$\begin{cases} \text{accept the offspring and go to 3} & \text{if } \delta(x) \vee \delta(y) = 1 \\ \text{go to 1} & \text{if } \delta(x) \vee \delta(y) = 0 \end{cases}$$

3. The new population includes the offspring from $x \otimes y$. Stop.

5 CIA crossover imposing alleles

The characteristic doing more attractive a genetic algorithm is the *implicit parallelism* property, in a simple way the algorithm works in the search space among schemata that try to be more and more better, although there are not guarantees of that when we apply the crossover or mutation operator a good scheme is destroyed and occurring that we lose a quantity potentially good of individuals. In this case, we try to determine a strategy that accompanies to the crossover designed to avoid the destruction possibility of a good scheme that we desire to keep in next generation.

We present another crossover operator to accelerate the exchange of good genes through evaluation statistic of the alleles from the better individuals of the population. This operator imposes to the offspring the most probable good material of the best individuals, versus traditional recombination from the genes of the parents. We propose make a *"pattern"* (a very good scheme), this pattern defines a set of *classes* and a *order relation*. This crossover operates so that the best class present in each one of the parents will be present in the offspring. We have titled this procedure: *imposing alleles to the offsprings.*

Construction of the pattern We propose an easy way to construct the *pattern*, consisting of fixing those alleles which are present in all the m best individuals.

5.1 Imposing alleles in the Uniform Crossover

It is checked if the *pattern* appears in an individual, in affirmative case high crossover probabilities are assigned to the wild cards and the low ones are assigned to the scheme fixed bit.

The simplest crossover probability that we can define is:

$$\begin{cases} p_c^i = 0 & \text{if the } i \text{ allocation is fixed in the } \textit{pattern} \\ p_c^i = 1 & \text{if the } i \text{ allocation isn't fixed in the } \textit{pattern} \end{cases}$$

5.2 CIA crossover in the framework of the class and relations

Now, we will define the proposed strategy in the framework of the relations and classes, which are defined by:

$$xR_{i_1,\ldots,i_j} \Leftrightarrow b_x(i_1) = b_y(i_1) \text{ AND } b_x(i_2) = b_y(i_2) \text{ AND } \ldots b_x(i_j) = b_y(i_j) \; j \in \{0,1\}$$

The objective is to decide who is the better class. For it that we will construct an order relation among all the classes using the *pattern class* C_0, which is a very good class that we have detected into the current population.

The order relation among the classes is established as:

$$C_i \succ C_j \Leftrightarrow d(C_i, C_0) > d(C_j, C_0)$$

being: $d(C_i, C_0)$="number of coincidence in the fix locations between classes C_i and C_0".

According to the established order we select two parents and take the another one belonging to the greatest class. The strategy is to make the crossover maintaining some child in that class or in another better to her. The objective is to preserve or to improve this class in such a way that we don't loss it in the next generation. The procedure is summarised by:

1. Fix the set of the class C_i verifiying $d(C_i, C_0) \geq 1$, and to form:
$$A = \{C_i / \delta_i(x) \vee \delta_i(y) = 1\}$$
Being $\delta_i(x) = \begin{cases} 1 & \text{if } x \in C_i \\ 0 & \text{if } x \notin C_i \end{cases}$

2. Take C_* such that $C_* \succ C_i$ for each C_i belonging A

3. Define the crossover $x \otimes y = \begin{cases} x_c \\ y_c \end{cases}$ verifying $\delta_*(x_c) \vee \delta_*(y_c) = 1$. To do that, we consider the crossover operator defined above.

6 A selection operator for optimisation problems (SOOP)

One of the characteristics that makes a genetic algorithm more attractive is that works with fitness function without imposing any restrictions. However when we know that the function has derivate we maybe losing useful information. We are obviating that the optimum (local or global) are points null derivate. We try here to increase the selection probability in the individuals that have null or almost null derivate. Furthermore it would be good for us that this subject weights more in the last generations of the algorithm. After of taking into account all that, we will modify the selection probability incorporating some defined weights, and in such a way that when the nearer to 0 in their derivate, the nearer to 1 will be to correspond the weight.

Therefore, the proposed selection probability for the individual x_i will be:

$$P(x_i) = \frac{f(x_i) e^{\frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}} \frac{t}{t+1}}}{\sum_{i=1}^{N} f(x_i) e^{\frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}} \frac{t}{t+1}}}.$$

7 Experimental results

We have considered in all experiments *steady-state* without duplicates with only two children to breed per iteration of the genetic algorithm (it corresponds to only two evaluations of the fitness function for each new population). All results presented are the average over independent 30 runs considering 90 individuals in the populations. We have considered as test functions the two following ones:

$$\frac{\sin(x)}{x} e^{\sin(x)}$$

in different intervals. It have been tested for the proposals of Approximated Confidence Interval (ACI), Crossover Imposing Alleles (CIA) and Modified Selection. The results are shown in Figures 1–3.

As conclusions we can observe from results that performance improve until the solutions are very near to the optimum. The different strategies can be used simultaneously by mix

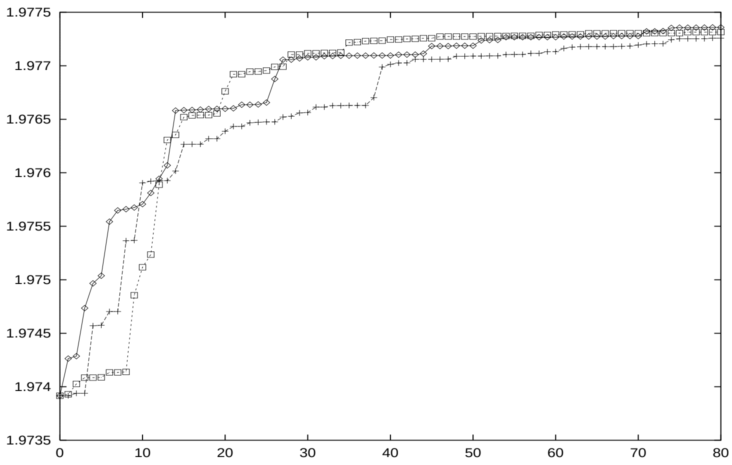

Figure 1: Function: $\frac{\sin(x)}{x}e^{\sin(x)}$. Interval: (-2.5,2.5). Two Points Crossover.

$\diamond$ ACI employing the best 7 individuals with $k = 3 * 1.64$
$+$ Without ACI
$\square$ ACI employing the best 30 individuals with $k = 4 * 1.64$

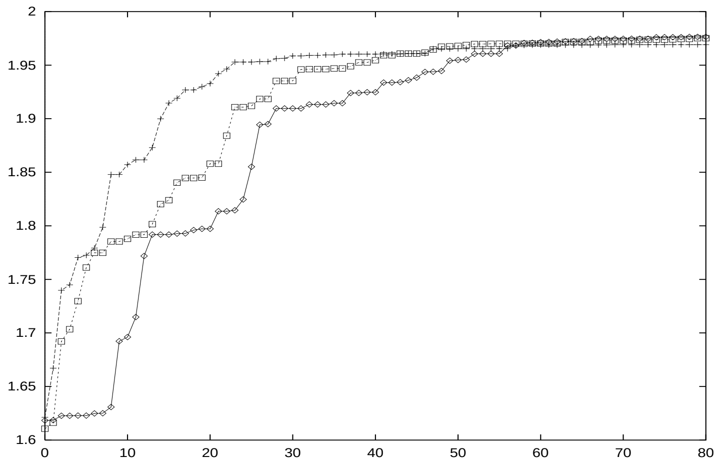

Figure 2: Function: $\frac{\sin(x)}{x}e^{\sin(x)}$. Interval: (0,100). Pattern made with the five best individuals.

$\diamond$ The best uniform crossover between gray code or binary code
$+$ Crossover Imposing Alleles(CIA) employing binary code
$\square$ CIA employing gray code

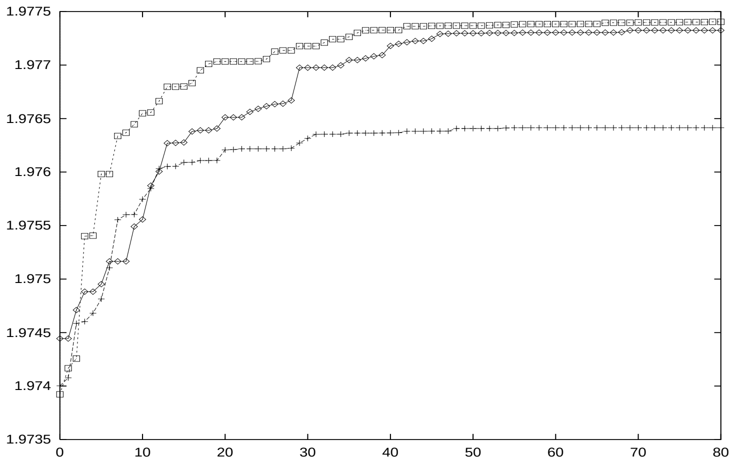

Figure 3: Function: $\frac{\sin(x)}{x}e^{\sin(x)}$. Interval: (-2.5,2.5). Uniform Crossover.

$\diamond$ Ranking Selection
$+$ Proportional Selection
$\square$ New Selection with decreasing factor $\frac{1}{20}$

all of them and in the future we hope to test this idea. Also is interesting to use only the proposals in a percentage of the first iterations of the simple GA to avoid the risk of the premature convergence problem.

8 Summary

We extract the following conclusions about the two problems that we have been presented in this work:

1. The methodology that we have followed, using GA, is a alternative to the classical cluster analysis; our proposal permit to assign any national council to the different groups, that we have obtained like best classification in interest groups. For this reason we have setlted the groups that we need to determine.

2. In this paper we have highlited some characteristics of the GAs to focus several ideas that we propose to accelerate the convergence of simple GAs. The main conclussion is that all proposals have had a common behaviour viewed from the experiments done, which is to increase velocity of the GAs very fast toward solutions very close to the solution of the problem.

References

[1] Davis, L., *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, 1991.

[2] Haupt, L. R. and Haupt, S.E., *Practical Genetic Algorithms.* John Wiley and Sons, New York, 1998.

[3] Jain, A. K. and Dubes, R.C., *Algorithms for Clustering Data.* Prentice-Hall, London, 1988.

[4] Kargupta, H. and Goldberg, D.E., SEARCH, Blackbox Optimization, And Sample Complexity. *FOUNDATIONS OF GENETIC ALGORITHMS. 4*, 291–324. Editors: Belew, R.K. and Vose, M.D., Morgan Kaufmann, 1997.

[5] Syswerda, G., Uniform Crossover in Genetic Algorithms, *Proceedings of the Third International Conference on Genetic Algorithms and their Applications*, 2–9, Morgan Kaufmann, 1989.

[6] Winter, G., Periaux, J., Galán, M., Mantel, B. and Sánchez, I., An Introduction on Global Optimization by Genetic Algorithms. *Algorithms for large scale systems. State of the Art and Applications in Sciences and Engineering.* Editors: Winter Althaus, G. and Spedicato, E., Kluwer, 1998.