Transactions on Information and Communications Technologies vol 19 © 1998 WIT Press, www.witpress.com, ISSN 1743-3517

Learning an optimized classification system from a data base of time series patterns using genetic algorithms

C. M. N. A. Pereira, R. Schirru & A. S. Martinez
Programa de Engenharia Nuclear/COPPE, Universidade Federal do Rio de Janeiro, caixa postal 68509, Rio de Janeiro - RJ, Brasil
EMail: cmnap@cnen.gov.br

Abstract

This work presents a novel methodology for pattern recognition that uses genetic learning to get an optimized classification system. Each class is represented by several time series in a data base. The idea is to find clusters in the set of the training patterns of each class so that their centroids can distinguish the classes with a minimum of misclassifications. Due to the high level of difficulty found in this optimization problem and the poor prior knowledge about the patterns domain, a model based on genetic algorithm is proposed to extract this knowledge, searching for the minimum number of subclasses that leads to a maximum correctness in the classification. The goal of this model is to find how many and which are the clusters to consider. To validate the methodology, reference problems, where the best solution is well-known, are proposed. Extending the scope of the application, the methodology is applied to a real problem, in which it is required to distinguish three nuclear accidents that may occur in a nuclear power plant. The misclassification rate was 5% in a total of 180 trials. To ratify the results an artificial neural network was designed and trained to solve the same problem. The results and comparisons are shown and commented.

1 Introduction

Due to the wide range of application, the pattern recognition has been intensely studied and improved. Many techniques such as classifier systems, e.g. in Goldberg[1], fuzzy systems, Kim[2] and Kosko[3], and neural networks, Kosko[3] and Steven[4], have been explored. This work has as its main objective to present an optimized method of using centroids applied to pattern recognition.

To introduce our methodology, it is firstly presented a the simplest classification method based on centroids, which we call the Simple Centroid Method (SCM). This method proposes the creation of one cluster per class, each one represented by their respective centroids. So, a given pattern is said to belong to the class which the centroid is the closest one. Each component of the centroid is given by the arithmetic average of the components of the training patterns that make part of the class. Let $C = \{a_1, a_2, \ldots, a_m\}$, where $a_j = (a_{j1}, a_{j2}, \ldots, a_{jn})$ be a class. The centroid c of the class is C is give by:

$$\vec{c} = \left(\frac{a_{11} + a_{21} + \ldots a_{m1}}{m}, \frac{a_{12} + a_{22} + \ldots a_{m2}}{m}, \ldots, \frac{a_{1n} + a_{2n} + \ldots a_{mn}}{m} \right) \qquad (1)$$

And the distance from one point to the centroid is given by the Euclidean distance between two points.

$$d(\vec{c}, \vec{a}) = \left[\sum_{i=1}^{n} \left(c_i - a_i \right)^2 \right]^{1/2} \qquad (2)$$

Figure 1 exemplifies graphically the pattern clustering and classification, based on the SCM for two classes represented by a set of points in a 2-dimensional space.

In the example showed in Figure 1, the patterns p_1 e p_2 are easily distinguished by their centroids. However, in the time series, the patterns are spread along the time axis, and they can often interlace themselves creating areas (time ranges) of confusion between the patterns, that may lead to misclassifications, as shown in Figure 2.

It is clearly observed that the application of the SCM to classify V_1 and V_2 would be catastrophic.

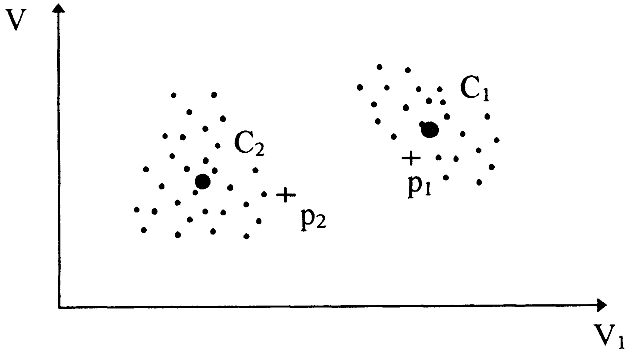

Figure 1: Example of two classes and their respective clusters C_1 e C_2 where the pattern p_1 is classified as belonging to class C_1 and p_2 as belonging to class C_2.

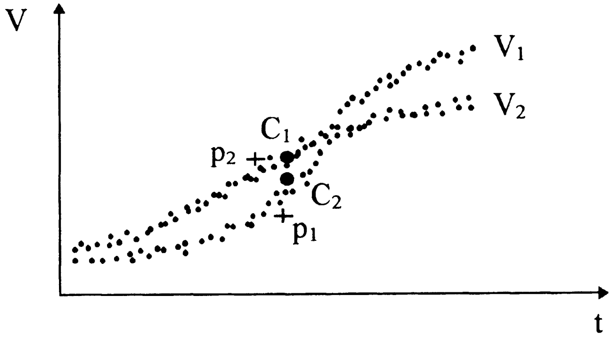

Figure 2: Example of two time series $V_1(t)$ e $V_2(t)$ and their respective centroids C_1 e C_2 where pattern p_1 is classified as belonging to class C_2 and p_2 as belonging to class C_1, causing misclassifications.

Consider now the hypothetical example of two other time series in Figure 3. Making use of the SCM it will be found that the classes centroids will be located at the middle of the time interval, that may lead to misclassification for all patterns to be classified.

Note that the example in Figure 3 presents classes that are easy to be distinguished visually or by simple rules. However, because of its symmetry this distinction becomes a difficult task to the SCM, that will have coincident centroids for both classes.

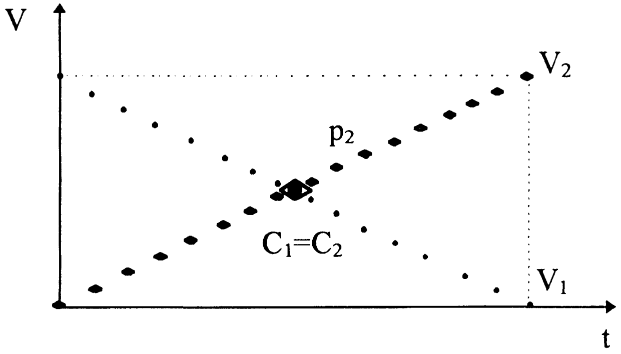

Figure 3: Two hypothetical time series where the centroids are coincident.

Now, consider that each pattern in Figure 3 have two subclasses (two clusters) and the partition is made at the center of the interval, as shown in Figure 4.

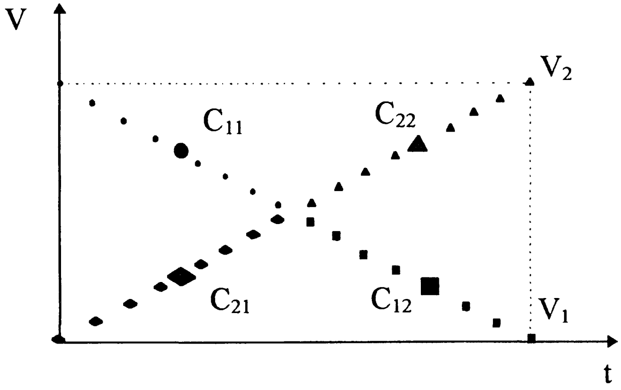

Figure 4: Two hypothetical time series partitioned into two subclasses each one.

Based on Figure 4, the simple algorithm described in Figure 5 may classify perfectly a given pattern P.

```
MinimumDistance=d(P,C(1,1));
For c=1 to NumberOfClasses
    For i=1 to NumberOfSubclassesPerClass
        If d(P,C(c,i)) < MinimumDistance Then
            MinimumDistance= d(P,C(c,i));
            Class=c;
        End If;
    End For;
End For;
```

Figure 5: The algorithm of classification on a multiple-centroid scenario.

In Figure 5, *C(c,i)* is the *i*th centroid of class *c* and *d(P,C(c,i))* is the distance between point P and the cluster *C(c,i)*.

In the example in Figure 4, it is easy to find the best centroid set that better classifies V_1 and V_2, but if the number of variables that characterizes each pattern is high, the visualization of the best centroids may be very difficult or in some case almost impossible. In this case, the distinction of the patterns may become a very difficult task. That's why it is necessary to use an optimization technique. In this work it is proposed the use of genetic algorithm, e.g. Goldberg[1] and Davis[5], that is a blind and global search technique (Gray[6] and Renders[7]).

2 The Minimum Centroids Set Method

The Minimum Centroid Set (MCS) method consists in finding the best set of centroids that better distinguishes the classes from each other, considering one or more centroids per classes. In other words, the MCS method intends to find the minimum number of time partitions (and the positions) of the classes in which the subclass centroid better classify a set of test patterns, using the simple algorithm in Figure 5, that we will call the MCS algorithm.

The concept of similarity for patterns that are represented by a great number of variable time series does not have any chance of being aided by visual perception. Hence, the mathematical foundations and their different kind of metrics comes to help in these cases. However, the simple methods may not be applied successfully in most real cases. But if the complex real problem is divided into simple small ones, it may be possible to apply the simple methods to each simple problem. The problem is how to find this minimum set of small problems.

The goal of the proposed method is to apply the well-known and simplest mathematical metric in the regions of the domain where they work well. But to find this sub-domains in which the application of the simple method is well-succeeded may be a np-hard or np-complete problem. Because of the nature of the optimization to be done and the poor prior knowledge about the search space, it is proposed the use of Genetic Algorithm (GA) in the optimization model.

The use of the genetic algorithm in the search for the best centroids set is the main subject of this work. Figure 6 shows a schematic diagram of the MCS method.

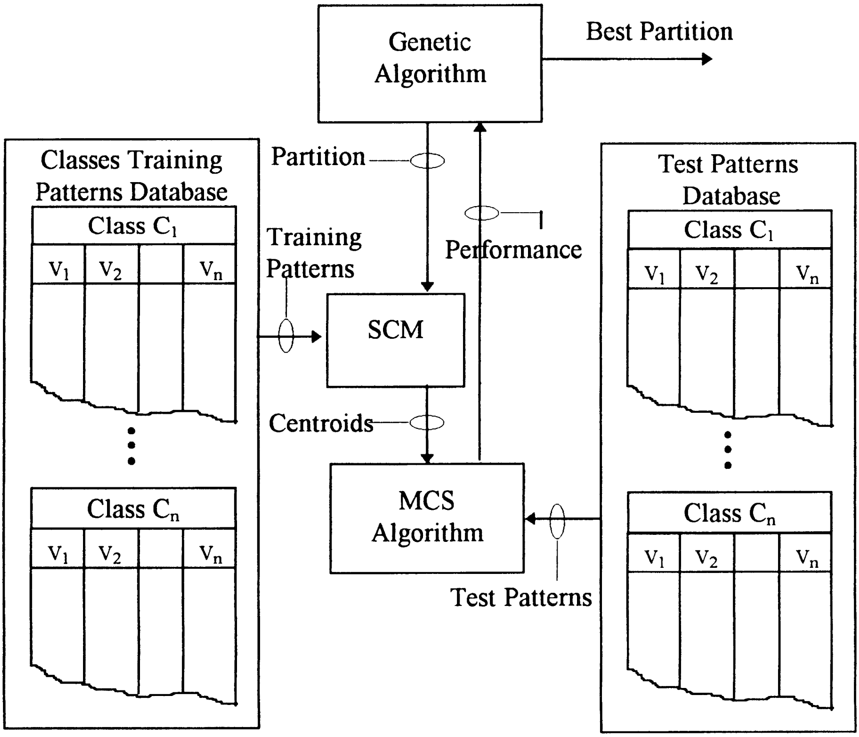

Figure 6: Schematic Diagram of the MCS method

The genetic algorithm generate candidates for the best partitions in which centroids (of each subclass) are calculated by the SCM, that provide the centroids to the MCS algorithm. The MCS algorithm tries to classify the pattern test set, sending the number of well succeeded classifications (performance) back to the GA that, in its turn, uses it to guide the optimization search. The process is repeated until either the convergence criteria is achieved or the performance is satisfactory, when the loop may be interrupted.

3 The Genetic Algorithm Model

Based on Darwin's[8] theory of species evolution, that involves natural selection processes, such as reproduction, crossover, mutation, and others, the Holland's[9] genetic algorithms manipulate a population of symbolic structures, that represent points in the search space, in order to evolve this population to its best adaptation, leading to the best solution for the problem. In the GA (Goldberg[1] and Davis[5]) the parameters to be varied in the optimization process are codified in a symbolic structure, metaphorically called genotype, that is formed by a set of genes that carry intrinsic characteristics of the symbolic individual. These characteristics dictate the adaptability of the individual in the environment, in which it may survive or die. The selection and evolution are made in such a way that stronger individuals have more chance to be selected, passing their characteristics to the offsprings. Making this way, from generation to generation, the tendency is that strong individuals become stronger and more numerous while the weak individuals may be extinct.

The GA starts the adaptation process from a set of possible configurations - a random generated population of individuals. Evaluating independently each individual by an objective function, the GA assigns to each one a fitness that predicts its resistance and adaptability. Then it is simulated a natural selection process in which the selection probability of a given individual is a function of the fitness.

The crossover in a classic GA is simulated by choosing a random cross point over the binary string that represents the genotype, followed by the change of parts between the two parents, as illustrated in Figure 7.

The mutation is simulated by the inversion of one of the bits of the genotype randomically chosen.

Taking into account the reproduction, crossover and mutation, it can be proved statistically, by the schemata theory, described in Goldberg[1], that strong individuals may be more numerous in subsequent generations, and the population converge to the optimum adaptation.

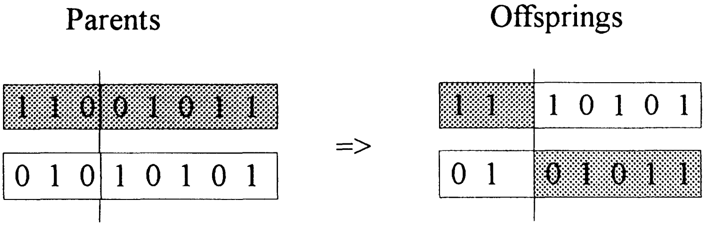

Figure 7: An example of crossover in a classic GA.

In this model, the clusters are coded into binary strings that have the length equal to the number of training patterns per accident. The groups of 1's (one) or 0's (zeros) together form a cluster or subclass. An example of the genotype that code the clusters of two hypothetical classes in Figure 8 is shown in Figure 9.

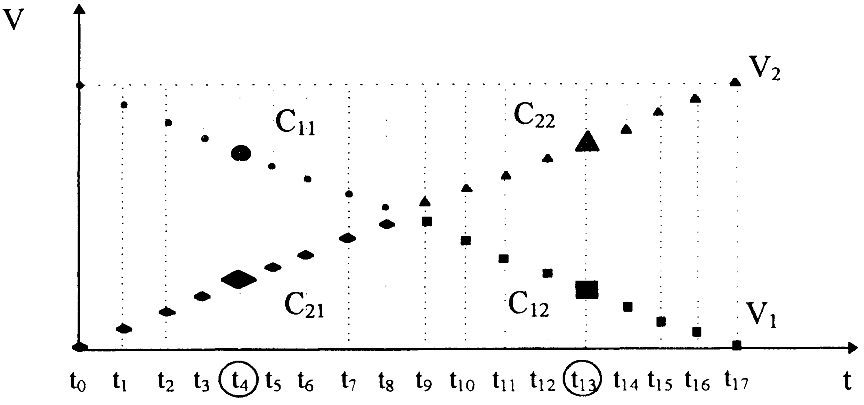

Figure 8: Two hypothetical time series.

Structures like the one shown in Figure 9 are over which the GA works to guide the optimization based on the objective function or fitness.

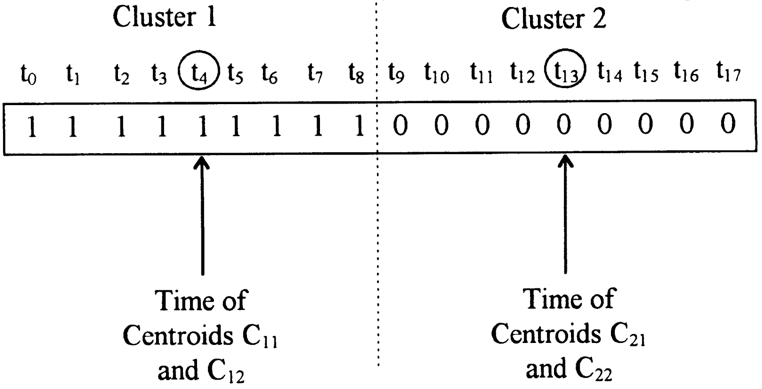
Cluster 1
Cluster 2
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11} t_{12} t_{13} t_{14} t_{15} t_{16} t_{17}
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
Time of Centroids C_{11} and C_{12}
Time of Centroids C_{21} and C_{22}

Figure 9: Example of genotype representing the clusters

The fitness reward the high performance of correct classification and low number of centroids, as described below.

$$f = K_d \,.\, P - (K_c + K_o) \,.\, C \tag{3}$$

where P is the performace (number of correct diagnosis), C is number of clusters, K_d is the weighting factor for the performance, K_c is the weighting factor for the number clusters and K_o is the offset factor.

The fitness suggested presents three constants to be adjusted according to the relative importance of the variables to be optimized.

In this work, it was used $K_d = 4$ and $K_c = 1$. This means that the performance is four times more important than the number of clusters. The value of K_o was 0.01 to impose a conflict resolution in cases where the variables are different and the fitness is the same (of course if K_o were zero). This value should be sufficiently small to distinguish combinations with the required precision.

For example, take $K_d = 4$, $K_c = 1$ and $K_o = 0$. If $P = 98$ and $C = 1$, then the fitness, f, is 391 ($f = 4 \times 98 - 1$), on the other hand, if $P = 99$ and $C = 5$, the fitness should be also 391 ($f = 4 \times 99 - 5$). To solve this conflict, it is necessary to establish who must win this competition. If K_o is set to 0.01, the fitness values would be 390.99 ($f = 4 \times 98 - 1.01$) and 390.95 ($f = 4 \times 99 - 5.05$) respectively, and the performance is said to be more important.

Once defined the genotype shape, and the fitness function, the last thing to do is to adjust the genetic parameters with the aim of optimizing the convergence process of the genetic algorithm.

4 Results

The procedure to validate the MCS method for pattern recognition consisted on two distinct fases. In the first one, three simple reference cases were created, in order to validate the precision of the proposed method, once the results (best centroids) are well-known.

It was proposed 3 scenarios including two classes represented by one variable time series each one, as shown in figure 10. The curves that appear in Figure 10 seems to be continuous, but they are not. They are discretized into 57 training patterns for each class generated using steps of 1 second.

The sample cases are very simple and the minimum centroids set can be easily identified by a simple look. But the GA is blind, fact that make these scenarios a very good test of the precision of the GA model.

As the GA works only with the time axis, to increase the number of variable time series that represents each class not necessarily increase the complexity for the GA, but to our vision.

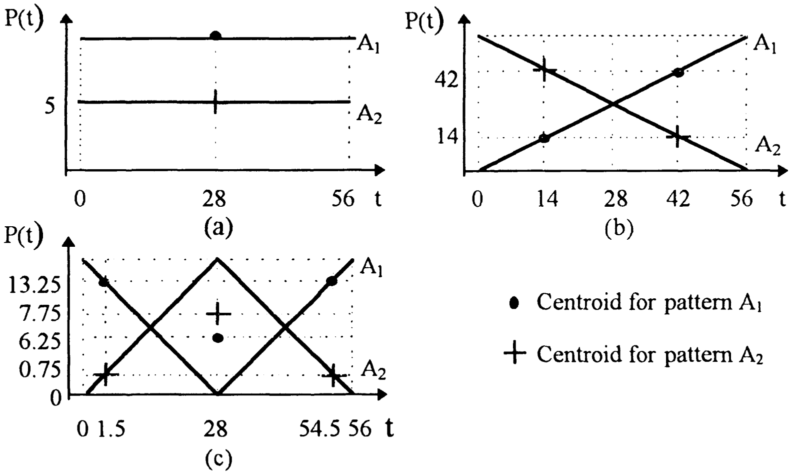

Figure 10: The test scenarios. (a) Test-1 (b) Test-2 (c) Test-3

The Genesis code, by Grefenstette[10], was used to implement the system, and the results obtained for the centroids optimization are shown in Table 1

Table 1: Best centroids obtained by the GA model

	Time of Centroid 0		Time of Centroid 1		Time of Centroid 2		Performance
	Theoretical	GA	Theoretical	GA	Theoretical	GA	Correct / trials
Test-1	14.0	14.0	42.0	42.5	-	-	112/112
Test-2	28.0	28.0	-	-	-	-	114/114
Test-3	1.5	3.0	28.0	28.5	54.5	53.5	108/110

It is observed that the optimum values for the centroids were reached with small errors, presenting very good diagnosis results (zero or very low percentage of misclassifications). The discrepancy between the result of the search and the optimum values is a characteristic of problems solved by Genetic Algorithms methods that have the aim to approximate the solution with a low computational cost. Not different from other Artificial Intelligence methods, the Genetic Algorithm is better applied when a formal method to solve the problem is very time consuming or non-existent.

Because of the good results obtained with the application of the method to the hypothetical scenarios a second phase of the validation test was proposed. The scope of the application was extended to a real case involving three transients in a nuclear power plant (NPP). Such kind of pattern recognition have been treated using artificial neural networks, e.g. Bartlett[11] and Bartal[12]. It was chosen three typical transients in nuclear power plants represented by 15 variables each, and it is assumed by hypothesis that all variables are required to the transient recognition

The transients considered were (i) the Black-out, (ii) the Lost Of Coolant Accident (LOCA) and (iii) the Steam Line Break. These transients were represented by tables with 60 points per variable, generated by simulation with time-step of one second (by hypothesis the transients can be characterized by time series of 60 seconds). The variables that considered in the nuclear transients are the Primary flux, .Nuclear power, Thermal power, Cold Leg temperature, Hot leg temperature, Average temperature, Subcoolling margin, Pressurizer pressure, Steam generator wide range, Steam generator narrow range, Steam pressure, Feed water flow, Break flow, Pressurizer level and Steam flow

The results obtained for the NPP real transients diagnosis are shown in Table 2. They are compared the SCM, the MCS and an Adaptative Vector Quantization (AVQ) Neural Network, e.g. Alvarenga[13] (NN), that were submitted to the same tests.

Table 2: Results for the real nuclear transient diagnosis problem

	Correct diagnosis / Total trials	Correct diagnosis (%)	Partitions
SCM	144 / 180	82%	1
MCS	171 / 180	95%	3
AVQ NN	155 / 180	86%	-

5 Conclusions

The search for the minimum number of centroids implies in getting a better statistic (the number of points per cluster is higher) that will lead to a more generalized representation. On the other hand, there is a compromise with the performance, that is the main objective of the pattern recognition system. The GA model proposed could efficiently handle the problem considering this compromise.

As it can be seen in Table 1 the GA was very precise to optimize the centroids for the reference scenarios, converging to a configuration that is very near to the known best solution. Table 2 shows the ability to optimize the relation between performance and number of centroids for the real nuclear transient identification, improving considerably the efficiency of the SCM. Besides, the results were better than the ones obtained by the AVQ NN. The fact is that the AVQ NN tries to learn the best vectors that distinguish the patterns, that is something analogue to finding simple clusters. Maybe that's why the result are not far from those obtained by the SCM.

Not differently from some kinds of NN, the learning process can be easily modified to allow "don't know" classification, considering limited action zones around the centroids. Another feature is that the learning process can also be made including some distortions in the training patterns. An advantage to be considered is that the learning time does not depend on the number of variable, as occurs with the NN learning.

Nowadays it is under development the prototype of a system which the main inputs are the training and test sets of pattern and the output is the program code based on the MCS algorithm shown in Figure 5. This program that generates programs may be useful in the design of MCS based pattern recognition systems.

References

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

[2] Kim, Y. S., and Mitras, S., An Adaptive Integrated Fuzzy Clustering Model for Pattern Recognition, *Fuzzy Sets and Systems*, **65**, pp. 297-310, 1994.

[3] Kosko, B., *Neural Networks and Fuzzy Systems*, Prentice Hall, 1992

[4] Steven K. Rogers and Matthew Kabrisky, *An Introduction to Biological and Artificial Neural Networks for Pattern Recognition*, Spie Optical Engineering Press, 1991.

[5] Davis, L., *Handbook of Genetic Algorithms*, VNR, New York, 1991

[6] Gray, P., Hart, W., Painton, L., Phillips, C., Trahan, M., Wagner, J., *A Survey of Global Optimization Methods, Sandia National Laboratories*, Albuquerque, 1997.

[7] Renders, J. M., Flasse, S. P., Verstraete, M. M. and Nordwik, J. P., *A Comparative Study of Optimization Methods for the Retrieval of Quantitative Information from Satellite Data*, EUR 14851 EN, Joint Research Centre, 1992.

[8] Darwin, C., The *Origin of Species by Means of Natural Selection*, John Murray, London, 1859.

[9] Holland, J. H., *Adaptation in Natural and Artificial Systems*, An Arbor, University of Michigan, 1975.

[10] Grefenstette, J. J., *A User's Guide to Genesis Version 5.0*, 1990.

[11] Bartlett, E. B. and Uhrig, R. E., Nuclear Power Plant Status Diagnostics Using an Artificial Neural Network, *Nuclear Technology*, **97**, pp. 272-281, 1992.

[12] Bartal, Y., Lin, J. and Uhrig, R. E., Nuclear Power Plant Transient Diagnostics Using Artificial Neural Network that Allow "Don't-Know" Classification, *Nuclear Technology*, **110**, pp. 436-449, 1995.

[13] Alvarenga, M. A. B., Martinez, A. S. and Schirru, R., Adaptive Vector Quantization Optimized by Genetic Algorithms for Real-Time Diagnosis through Fuzzy Sets, *Nuclear Technology*, **120**, 3, pp. 188-197, 1997.