Real Coded Genetic Algorithm for Systems Modelling

Rafael del Castillo Gomariz, Sebastián Ventura Soto, Eloy Rafael Sanz Tapia, César Hervás Martínez

Department of Computer Science and Numerical Analysis. University of Córdoba. Rabanales Campus. "Einstein" building. 14071-CÓRDOBA. Spain.
rcastillo@uco.es, sventura@uco.es, ersanz@uco.es, chervas@uco.es

Abstract. In this work a proposal is made to use real-coded genetic algorithms for systems modelling, in situations where a model based on response surfaces can be found. An algorithm is proposed which can select a minimum dimensionality model, identifying the analytical model and improving its generalization capabilities. This algorithm uses a double coding (real and binary), uses adapted specific real-coding operators and its aptitude function includes terms considering the sum of squared residuals and the number of coefficients. The new method has been tested by applying it to a synthetic model and a real-world model (kinetic analysis). The results are promising: errors are comparable to an artificial neural network model, and the model is much less complex and more interpretable.

1 Introduction.

1.1 Real-coded genetic algorithms.

Genetic algorithms (GAs) are stochastic optimization methods which perform a blind, multiple and general purpose search for quasi-optimal solutions. They maintain a population representing a set of possible solutions. This population undergoes a biased selection process which favours the best candidates. It also suffers certain transformations whose objective is to find new candidate solutions. The selection has two motivations: in the short term the best individuals have more chance of surviving, and in the long term they have more probabilities of obtaining descendants.

GAs do not work directly on the problem's domain, but on representations of the problem's elements in the form of binary, integer of real strings. Each possible representative string is called an individual, and it is divided into one or more chromosomes. These chromosomes are composed of genes and the genes are formed by alleles. Alleles are the atomic elements of the string.

The aforementioned transformations are carried out by the crossover and mutation operators. They establish an adequate equilibrium between exploration and exploitation. This makes GAs very popular for the resolution of a wide variety of highly complex combinatorial problems and engineering problems with difficult restrictions, which can only be solved by approximating their optimal values.

The mutation operator randomly modifies one or more alleles from a given chromosome with a given probability, thus increasing the structural diversity of the population. It is an operator clearly oriented to exploration, restoring the genetic material lost during the selection phase and exploring new solutions to prevent the algorithm from a premature convergence to a local optimum. This way, the probability of reaching a given point inside the search space is assured to be nonzero.

The crossover operator combines the characteristics of two or more parent individuals to generate better offspring. The idea is based on the interchange of information between good chromosomes to obtain better descendants. This way, the crossover operator implements a depth search or exploitation, leaving the breadth search, or exploration, to the mutation operator. This policy, which intuitively seems very natural, makes the population converge to inner values of the search space, thus producing a fast reduction of the diversity in the population, which could cause a premature convergence to a non-optimal solution.

The first studies suggested that binary coding was the most adequate [Goldberg91]; nevertheless later studies [Radcliffe92] show formally that no intrinsic advantages can be obtained from the selection of the alphabet for the strings. This makes possible to use other representations, more adequate to the problem at hand. One of the most important is real coding, which seems quite natural in optimization problems over continuous domains, where each gene represents one variable of the problem. Now the precision of the solution depends only on the computer system used for the simulation of the GA. The values of the genes are kept in the allowed intervals for their variables, so the operators have to take this restriction into account. This type of GAs are known as Real Coded Genetic Algorithms (RCGAs).

RCGAs can use crossover and mutation operators from binary GAs, but their performance is better when using operators specifically designed for this representation [Herrera98]. In the experiments for this work, Blx-α crossover, multiparent CIXL2 crossover and non-uniform mutation will be used.

1.2 Systems modelling.

Modelling of physicochemical phenomena is a extraordinarily interesting problem for researchers in a great number of application areas. In many fields the need arises of establishing a functional relationship between a phenomenon, expressed as a series of motivating variables, and a measure of the phenomenon's manifestation.

Typically, this problem has been tackled as a regression problem. The researcher, using her knowledge and experience, proposes one or more analytical models for the phenomenon and then applies regression techniques to adjust the parameters. This method is quite tedious, as it is not always possible to have a single model. Moreover, especially in high dimensionality, high nonlinearity models, it is very difficult to postulate an analytical model. In this kind of systems, artificial neural networks have given good results [Hervás99]. The problem with artificial neural networks is the difficulty in interpreting the produced model, as the researcher is normally interested in obtaining expressions that help to sustain the theories explaining the phenomena, and in the case of neural networks the expressions of the obtained models are not easy to interpret at all.

Approaching the modelling problem as a symbolic regression problem using genetic programming [Banzhaf98] is an interesting alternative as the resulting expression for the model is analytical and easily interpretable. Some examples of this kind of modelling can be seen in [Cordón99]. Nevertheless, this advantages are not always obtained as, in most cases, the symbolic regression problems are solved by genetic programming trying to minimize the error, and then the resulting expressions, though very well adjusted to the data, are difficult to interpret.

In this work the use of real-coded genetic algorithms [Michalewicz94] is proposed for the development of analytical models in situations where the family of functions to which the model belongs is known (in this case, response surfaces). Although the method used is a form of regression using genetic algorithms, it presents advantages compared to conventional regression. First, as the optimization is not based on any kind of gradient, non-differentiable equations can be used. But the most interesting characteristic of this method is that the aptitude function can contain terms to strengthen some properties of the model as, for example, the simplicity, expressed as the number of terms in the model. In this way, the analytical model of the phenomenon can be identified, giving the researcher valuable information.

The results obtained on the chosen problems have been very promising, compared to those previously reported [Hervás99] using conventional regression and artificial neural network models.

The rest of this work is organized in the following way: next section exposes the proposed methodology. In sections 'application' and 'experiments' some problems to which the method has been applied are presented. Last, in section 'results' the obtained results are presented and compared with those in the bibliography.

2 Proposed method.

In general, the modelling of a system whose equation is known is a conventional regression problem. In this type of problems there is a functional relationship between a group of independent variables x_i and a dependent variable y:

$$y = f(\beta_0, \beta_1, ..., \beta_m, x_0, x_1, ..., x_n)+\varepsilon \qquad (1)$$

where β_i are the coefficients to be adjusted so that the sum of squared residuals is minimized and ε is a term that represents other sources of variability not accounted for in f. Normally, ε is treated as a random variable with a normal distribution $N(0, \sigma^2)$. This optimization problem can be solved using a classic algorithm or a genetic algorithm. If the second solution is chosen, individuals would be coded by a set of genes, each one representing a coefficient. Figure 1 shows a typical individual in one of these problems, with the genes representing the equation's coefficients.

The use of a genetic algorithm presents some advantages inherent to the nature of this kind of

β_0	β_1	β_2	β_3	...	β_m

Fig. 1. Representation of an individual for regression problems using genetic algorithms.

algorithms. First, as the optimization does not use derivatives, the model is not restricted to differentiable functions, but can include any type of function, like step functions, functions with different definition domains, etc. Second, as the genetic algorithm optimization is guided by the aptitude function, this can include terms which take into account not only the estimation error, but also other factors as the generalization capability of the model (expressed as the complexity of the resulting expression).

2.1 Surface response models and genetic algorithms.

Usually, the *f* function (see equation (1)) is a first-order or second-order polynomial. This empirical model is called a response surface model (RSM) [Rawlings98][Myers02].

Surface response models explain a wide range of phenomena. The expression defining them is a grade *G* polynomial in each variable to be studied. The form of the functions, therefore, is:

$$f(x_1,x_2,\ldots,x_n) = c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{\substack{i,j=1 \\ i\le j}}^{n} c_{ij} x_i x_j + \sum_{\substack{i,j,k=1 \\ i\le j,j\le k}}^{n} c_{ijk} x_i x_j x_k + \ldots + \sum_{\substack{i_1,i_2,..i_G=1 \\ i_k\le i_{k+1}}}^{n} c_{i_1 i_2 .. i_G} x_{i_1} x_{i_2} \ldots x_{i_G} \quad (2)$$

Where *G* is the grade of the model, x_i the independent variables, *n* the number of independent variables and c_i the coefficients. Expression (2) is much more legible in the case of a quadratic response surface (*G*=2):

$$f(x_1,x_2,\ldots,x_n) = c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{i,j=1}^{n} c_{ij} x_i x_j \quad (3)$$

In many RSM problems it is convenient to transform the natural variables to typified variables which are standardized, that is, they have mean 0 and standard deviation 1.

When trying to model a phenomenon using the method proposed in this work, individuals will be coded with as many genes as coefficients in the intended model. This number is a function of the number of variables and the degree of the model. For instance, in the case of equation (3), our individuals would have *3n+1* genes, being *n* the number of independent variables. Figure 2 shows and individual representing a grade 2 response surface model with three independent variables.

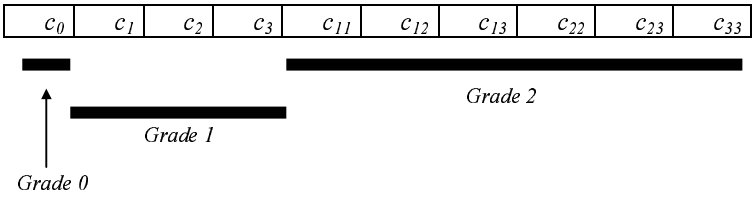


Fig 2. Individual representing a grade 2 response surface model with three independent variables.

As noted before, the easy interpretation of the models is a very desirable characteristic in any model. Equally less important is the requirement for the simplicity of the model. The simpler a model is, the easier it is to interpret, the fewer patterns are necessary to obtain its coefficients and, in general, the greater it is the generalization capacity. The proposed method allows the selection of the easiest possible model (the one with fewer terms), giving smaller errors. The coding of an individual is slightly different from the one depicted above. The individual has a gene for each coefficient in the model, but each gene has two well defined parts. One of them is an allele indicating the presence or absence of the monomial in the model. The other allele is used to store the value of the coefficient. Figure 3 presents an individual representing a grade 2 response surface model with 3 variables, adapted to this method.

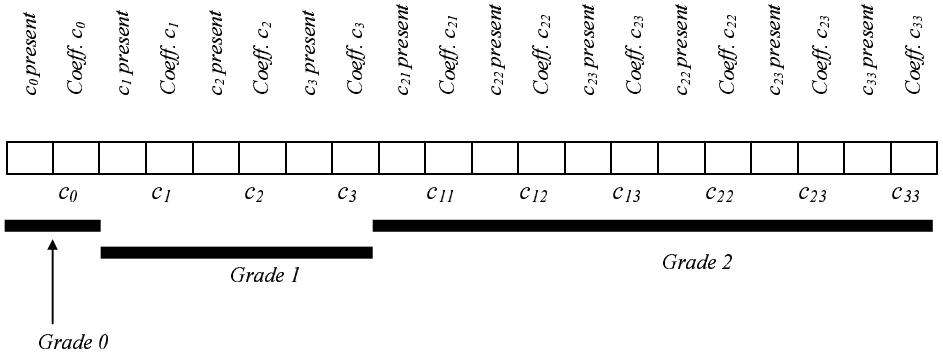


Fig 3. Individual modified to allow selection of the simplest model. Grade 2 response surface model with 3 independent variables

This representation schema does not select by itself expressions with a minimum number of terms, but the ones producing minimal error. To obtain the desired effect a term must be included in the aptitude function to reward simple models. This way, the problem becomes a two-objective problem: the error should be minimized and, on the other hand, it is interesting to obtain models with a small number of coefficients. As the number of objectives is reduced, the decision has been to create a single aptitude function which is a linear combination of them, weighting their importance through a coefficient. So, the aptitude function results:

$$A = (1-\alpha)A_{error} + \alpha A_{compl} \qquad (4)$$

where A_{error} represents the error term, A_{compl} represents the complexity term and α is the weighting coefficient.

The result of applying this algorithm is a minimum dimensionality model belonging to the family of functions in which the search was performed. The exact model does not need to be known, as the GA finds the best one inside the family.

3 Application.

The method has been applied to two experiments, the first one using synthetic data and the second one using data from a real kinetic modelling problem.

The reason for experimenting with synthetic data before using real data is the need to check whether the method was able to obtain the exact model of a phenomenon, or at least obtain a good approximation to the model, starting from a model with great amplitude, in which the original model is comprised. To check this point, an equation has been modified by adding noise and then samples from that function have been used to estimate the original model.

3.1 Synthetic Data.

A grade two polynomial with three independent variables (x, y, z) has been randomly generated. The expression is:

$$-9{,}281189855 - 0{,}371469568y - 6{,}635085107yz - 7{,}687441866z^2 \qquad (5)$$

Next, a set of learning patterns has been generated using this polynomial. Each pattern contains three independent variables (x, y, z) and dependent variable (result of equation (5)) with a random uniform error applied to it.

These data are similar to those obtained in many physicochemical phenomena with a certain level of noise in the sampling.

It has not been necessary to know that the model is a grade two polynomial, something that would be required when using any other conventional regression method, or even a normal genetic algorithm (not using the proposed method). For the experiments, the model and its coefficients have been approached using grade five response surfaces (polynomials with 56 terms) and grade ten response surfaces (polynomials with 286 terms), considered broad enough to start the search from.

3.2 Chemical kinetic.

This problem tries to solve species mixtures based on the perturbation on a oscillating chemical reaction, the Beluzov-Zhabotinski reaction [Jiménez98]. The goal is to model the relationship between the perturbation produced when a certain quantity of galic acid an pyrogalol mixture is added to the oscillating system and the concentration of each substance. The model is based on quadratic and cubic response surfaces. The data is a register of signal-time (independent variables) and the species concentrations (dependent variables). Figure 4 show a typical register obtained after three consecutive perturbations of the system. The upper right corner shows a magnified register of the perturbation. A pattern would contain n independent variables (the number of signals in the perturbation) and two dependent variables (concentrations of pyrogalol and galic acid).

Given the high dimensionality of the problem and the high correlation among the variables in the input set, a preprocessing of the data has been performed. A least squares regression has been applied to fit them to a gaussian curve, whose expression is:

$$S = a_m e^{-1/2(t_i - t_m)^2 / s^2} \tag{6}$$

where a_m is the maximum of the response curve, t_m is the instant in which that maximum is reached and s is a parameter associated to the signal dispersion. This preprocessing is justified, considering the mechanism of the reaction, in [Hervás99]. In that work it is also shown that a response surface over the variables a_m, s and t_m explains adequately the behaviour of the system. Therefore, two response surface models will be obtained. Both of them will have a_m, s and t_m as independent variables. The dependent variable of each model will be the concentration of pyrogalol and the concentration of galic acid, respectively.

4 Experimental section.

4.1 Pattern set.

A. Synthetic data.

The pattern set used to model the synthetic data comprises 90 patterns with random values of the independent variables in the interval [-100,100]. Each pattern is composed of three independent variables (x, y, z) and a dependent variable (result of equation (5)) with a random uniform error in the interval [-2.5%,2.5%] applied to it.. Two thirds of the patterns (60) are the training set and the rest (30) constitute the generalization set.

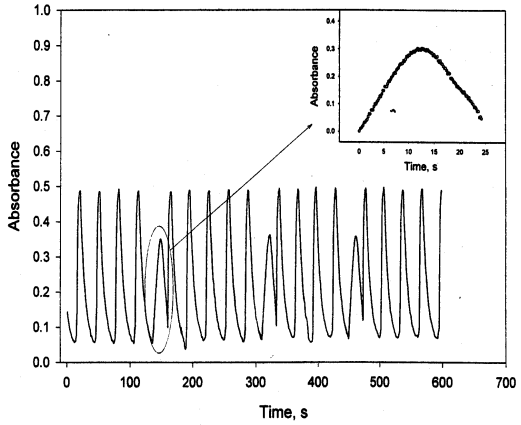


Fig. 4. Perturbation produced on Beluzov-Zhabotinski reaction when a mixture of galic acid and pyrogalog is added.

B. Chemical kinetic.

The pattern set if composed of 78 patterns, 26 synthetic samples triplicated, containing uniformly distributed concentrations of galic acid an pyrogalol. Of the three replicas, two of them chosen at random

form the training set and the other one is the generalization set. Each pattern is formed by three independent variables (the estimators of the gaussian curve fitting) and two dependent variables (pyrogalol and galic acid concentrations). Originally, the independent variables were the signal/time records, which were fitted to a gaussian curve.

4.2 Genetic algorithm.

Table 1 summarizes the parameters for the genetic algorithms used to model synthetic and chemical kinetic data.

ALGORITHM'S GENERAL ASPECTS			
Population size		500 individuals	
Operators	*Duplication*	P_d=0.2	Tournament selection
	Crossover	p_c=0.6	Tournament selection Blx (α=0.5) CIXL2 (n=5)
	Mutation	p_m=0.2	Random selection Non uniform mutation (parameter b=5)
Stop criterion		500 generations	

Table 1. Parameters used in the genetic algorithms for response surfaces modelling.

As noted before, the aptitude function has two terms. The first one represents the error (related to the minimization of the sum of squared residuals) and the second one represents the model's complexity (related to the minimization of the number of coefficients).

The first term (A_{error}) is a transformation of standard error of prediction (%SEP), an adimensional coefficient given by the following expression:

$$SEP = \frac{100}{\bar{y}} \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \qquad (7)$$

where y_i represents the function's value in that point, $\hat{y}_i$ is the estimated value and $\bar{y}$ is the average value of all the y_i.

The second term (A_{compl}) linearly modulates the number of terms in the expression, growing as the number of terms (n_T) decreases.

So, the aptitude function would be:

$$A = (1-\alpha)(1-\frac{SEP}{K}) + \alpha\left(1 - \frac{n_T - n_{Tm}}{n_{TM} - n_{Tm}}\right) \qquad (8)$$

Where coefficients n_{Tm} and n_{TM} represent, respectively, the minimum and maximum number of coefficients that the model is allowed to have, and constant K allows to modulate the SEP value to highlight the differences among patterns so as to obtain an equilibrium between both objectives.

This aptitude function is non-decreasing, with 1 as its minimum value. This value would be obtained if the standard error of prediction was zero and the model had n_{Tm} terms.

The number of genes for each individual in the population will depend on the grade of the response surface chosen for the search. Table 2 show this data for both experiments.

	RS grade	*# of coefficients*
Synthetic data	5	56
	10	286
Chemical kinetic	2	10
	3	20

Table 2. Number of coefficients and response surface grade for each experiment.

Each coefficient will be expressed by a gene, which in turn will consist of two alleles. The firs allele contains the real value of the coefficient and the second allele is a selector indicating the presence of absence of the monomial in the model. Conceptually, the method uses a hybrid genetic algorithm including both real and binary alleles, although the implementation uses a real-coded genetic algorithm in which selection alleles are real alleles in the interval [0, 1] which will be rounded: if the value is greater than or equal to 0.5, the coefficient will be selected.

Crossover operators used in the algorithms were Blx-α crossover [Eshelman93] and CIXL2 multiparent crossover [Hervás02]. For mutation, the non-uniform mutation operator has been used. These crossover operators, specific of real coding, have been adapted to work with the aforementioned double coding.

4.3 Implementation.

All the algorithms have been implemented in Java using version 1.3.1 of Sun Microsystems Java Development Kit, and the JCLEC class library for evolutionary computation [Ventura02].

Regression models used for comparison have been execute on the SPSS 8.0 statistics package, and the artificial neural network models have been created using a C tool developed by members of the AYRNA research group at the University of Córdoba.

All the experiments have been performed on a Pentium-III 800 MHz computer, under the Linux operating system.

5 Results.

For each experiment, 10 runs have been performed using the parameters described in the previous section.

5.1 Synthetic data.

As noted before, the model's coefficients have been located using grade 5 and grade 10 response surfaces to check whether the method was able to find the original model, from which the training patterns were taken. To test the acceptability of the results, first the error was calculated using the training and test patterns from the original model. As expected, the error was very low (2.328 %SEP for training data and 1.869 %SEP for test data).

After that, a model was obtained using the proposed method and the results obtained using the two versions of the algorithm (RS^5 and RS^{10}) show even less error (2.174 %SEP for training data and 1.591 %SEP for test data). The obtained model had three coefficients instead of the four in the original model, a very promising result.

The 10 runs of the genetic algorithm obtained the same model in less than 100 generations:

$$-10 - 6{,}571341yz - 7{,}678008z^2 \qquad (9)$$

If compared to equation (5), it can be seen how the three coefficients are very similar to those of the original model, although the coefficient for y has not been considered (in the original model was very small). It is also interesting to note how the obtained model does not include the independent variable x, which was in the training patterns but not in the original model.

If a classic regression method or a traditional genetic algorithm had been applied, values would have been obtained for all the coefficients (56 for RS^5 and 286 for RS^{10}) and the error would be similar, which is not useful for the researcher when she tries to obtain conclusions about the phenomenon.

Crossover operators have been Blx-α and multiparent CIXL2. The latter was only used for the RS^{10} version of the algorithm, as the Blx-α performance was bad in that case. The mutation operator has been non-uniform.

5.2 Chemical kinetic.

In the version of the algorithm using quadratic response surfaces, the best results show an error (expressed as %SEP) of 4.427 form pyrogalol and 10.438 for galic acid. Figures 5a and 5b show the existing correlation between the estimated values and the real ones. As can be seen, the slopes of the regression (close to 1) and the proximity to zero at the origin show very little deviation between real and predicted values. The correlation coefficient also shows the good fit between both data sets.

	Obtained expression		%SEP
PYROGALOL	$[Py] = -1{,}130 + 0{,}246s + 1{,}132a_m^2$	(RS^2)	4,426835
	$[Py] = 2{,}812 + 1{,}134a_{mt} + 1{,}997s_t - 2{,}013a_{mt}t_{mt}^2 + 2{,}719st_{mt}^2$	(RS^3)	3,787163
GALIC ACID	$[Ga] = 5{,}831 + 18{,}227a_m - 0{,}198s - 0{,}558t_m - 0{,}618as + 0{,}011t_m^2$	(RS^2)	10,438229
	$[Ga] = 3{,}232 + 6{,}618a_{mt} - 4{,}688s_t + 6{,}799a_{mt}t_{mt}^2 - 8{,}879st_{mt}^2$	(RS^3)	8,006216

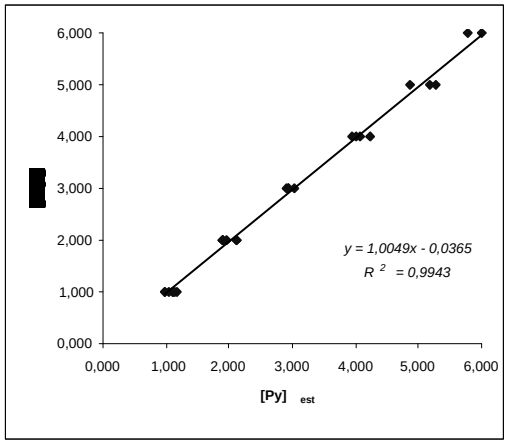
(independent variables of RS^3 models are preprocessed (typified))

Table 3. Best models obtained using the genetic algorithm with coefficient selection

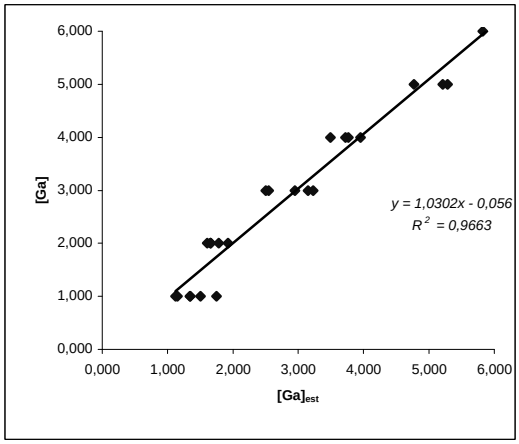
In the version using cubic response surfaces, slightly higher error values were obtained (%SEP=3.718 for pyrogalol and 8.006 for galic acid). Figures 5c and 5d show the existing correlation. The same comments apply. In this version, the learning patterns had to be tipified, as their variance was high.

In both cases the number of coefficients has been reduced from that of the starting model. Table 3 shows the expressions of the obtained models.

Regression models using *quadratic* response surfaces

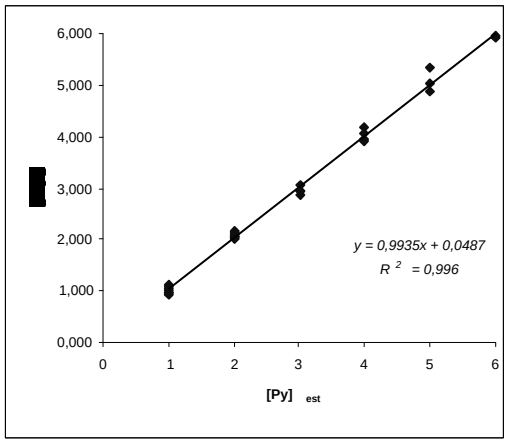


(a) Pyrogalol

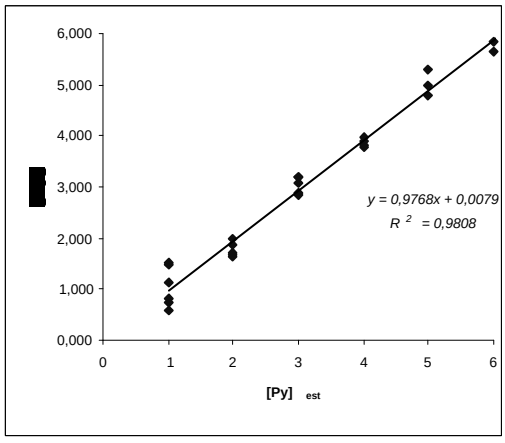


(b) Galic acid

Regression models using *cubic* response surfaces



(c) Pyrogalol



(d) Galic acid

Fig. 5. Graphs showing correlation between values predicted with the best models obtained and the concentrations of the studied species.

Table 4 shows the results obtained in the best regression models and also those obtained in [Hervás99] using a backpropagation neural network with architecture 3:3s:2l, two traditional genetic algorithms (with the single goal of minimizing the error) using quadratic and cubic response surfaces and those resulting

from a manual regression method based on trial and error: perform a regression with 10 coefficients (quadratic response surface), obtain the model, check coefficients close to zero, eliminate one of them, repeat the process until all coefficients are significantly different from zero. The models obtained for the pyrogalol and galic acid were the following:

$$[Py] = -1{,}660 + 1{,}357\,a_m + 0{,}264\,s \qquad (10)$$

$$[Ga] = 3{,}509 + 14{,}674\,a_m - 0{,}264\,s - 3{,}428\,a_m^2 \qquad (11)$$

Both expressions are quite similar to those obtained by the proposed method, using grade 2 response surfaces, although in the case of pyrogalol, our expression is quadratic on a_m and the former is linear. It is also important to notice that the expressions obtained using our method produce lower generalization errors than those obtained directly through regression. This proves that the genetic algorithm is selecting the terms of the most adequate polynomial to model the studied phenomenon, with the advantage of the automatization of the process.

Table 4 also shows that the results obtained using the neural network are comparable (and in some cases worse) than those obtained using our method, but the model is much more difficult to interpret, as it is a sum of three sigmoids whose arguments are linear combinations of the input data (expressions for that model can be seen in [Hervás99]). The same can be said from the results of a traditional genetic algorithm (optimizing the coefficients for a minimum error). 10 coefficients are obtained in the first case, and 20 in the second. Too many for a researcher to obtain conclusions.

	PYROGALOL			GALIC ÁCID		
	Training SEP	**Test SEP**	**Number of Coeff.**	**Training SEP**	**Test SEP**	**Number of Coeff.**
Neural network	3,790000	3,630000	18	9,870000	8,450000	18
Nonlinear regr.	5,254655	4,894726	3	13,218830	12,355038	4
GA (RS^2)	4,168634	3,658061	10	10,954901	9,385597	10
GA (RS^3)	3,651365	3,507542	20	9,299867	8,487273	20
GA with Coeff. Selection.	4,454962	4,426835	3	11,210121	10,438229	6
	3,692012	3,717863	5	9,352459	8,006216	6

Table 4. Comparison with other models.

6 Conclusions.

This work has demonstrated the ability of genetic algorithms to solve modelling problems in cases where there is a priori knowledge of the family of functions describing the phenomenon to investigate. An aptitude function has been proposed that, apart from considering the sum of squared residuals, also weights the simplicity of the model. This leads to expressions evolved to a minimum size, thus improving their interpretability and their generalization ability. This procedure has an advantage over statistics tests used to prune coefficients, much more tedious and, in some cases, influenced by the subjective appreciations of the researcher. It has been determined that using this algorithm the obtained results are better than those obtained using nonlinear regression (where the exact form of the model must be known), and comparable to those of artificial neural networks (where the interpretation of the model is more complex).

7 References.

[Banzhaf98] W. Banzhaf, P. Nordin, R. E. Keller, F. D. Francone.. *Genetic Programming - An Introduction* - Morgan Kaufman Publishers 1998.

[Cordón99] O. Cordón, F. Herrera and L. Sánchez. Some Electrical Distribution Problems Using Hybrid Evolutionary Data Análisis Techniques. *Applied Intelligence*, 10, 5-24. 1999.

[Eshelman93] Eshelman, L. J. and Schaffer, J. D. Real-coded genetic algorithms and interval-schemata. In Whitley, L. D., editor, *Foundation of Genetic Algorithms 2*, pages 187c3.3.7:1-c3.3.7:8.-202, San Mateo. Morgan Kaufmann

[Goldberg91] Goldberg, D. E. *Real*-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, 5(2), pp. 139-167, April 1991.

[Herrera98] Herrera, F., Lozano, M. and Verdegay, J. L. Tackling real-coded genetic algorithms: operators and tools for behavioural analysis. *Artificial Intelligence Review*, (12):265-319. Kluwer Academic Publishing. 1998.

[Hervás99] Hervás, C., Toledo R. and Silva, M. Use of Pruned Computational Neural Networks for Processing the Response of Oscillating Chemical Reactions with a View to Analyzing Nonlinear Multicomponent Mixtures. *Journal Chem. Inf and Comp. Sci.*, 41(4), 1083-1092, 1999.

[Hervás02] Hervás, C., Ortiz, D., García, N. Theoretical Analysis of the Confidence Interval based Crossover for Real Coded Genetic Algorithms. *Parallel Problem Solving from Nature 7*. 2002. Submitted.

[Jiménez98] Jiménez Prieto, R., Silva, M., Pérez Bendito, M. D. and Hervás, C. Approaching the use of oscillating reactions for analytical monitoring. *Analyst*,123, 1R-8R, 1998.

[Michalewicz94] Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag. 1994.

[Myers02] Myers Rayomond, H. M. and Montgomery, D. C. *Response surface methodology: process and product optimization using designed experiments.* Second Edition. John Wiley and Sons. 2002.

[Radcliffe92] Radcliffe, N. J., Non-Linear Genetic Representations. *Parallel problem solving from nature 2*, pp. 259-268, North-Holland, 1992.

[Rawlings98] Rawlings, J. O., Pantula, S. G., Dickey, D. *Applied Regression Analysis: a research tool*. Springer-Verlag.1998.

[Ventura02] Ventura, S., Ortíz, D. and Hervás, C. JCLEC. Una librería de clases Java para Computación Evolutiva. *Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados*. 2002.