Available online at www.sciencedirect.com
ScienceDirect

Chemometrics and Intelligent Laboratory Systems 85 (2007) 232–242

Chemometrics and intelligent laboratory systems

www.elsevier.com/locate/chemolab

Memetic algorithms-based artificial multiplicative neural models selection for resolving multi-component mixtures based on dynamic responses

César Hervás [a,1], Manuel Silva [b,*]

[a] Department of Computer Science, Albert Einstein Building, Rabanales Campus, University of Cordoba, E-14071 Cordoba, Spain
[b] Department of Analytical Chemistry, Marie-Curie Building (Annex), Rabanales Campus, University of Cordoba, E-14071 Cordoba, Spain

Received 7 November 2005; received in revised form 7 April 2006; accepted 29 June 2006
Available online 7 August 2006

Abstract

The potential of the product unit neural networks built by using different memetic evolutionary algorithms for the simultaneous determination of mixtures of analytes based on dynamic responses was investigated. For this purpose, three methodologies for obtaining the structure and weights of neural networks were proposed, based on the combination of the evolutionary programming algorithm, a clustering process and a local improvement procedure carried out by the Levenberg–Marquardt algorithm. To test these approaches, two phenol derivatives, pyrogallol and gallic acid, were quantified in mixtures based on their perturbation effect in a classical oscillating chemical system, namely, the Belousov–Zhabotinskyi reaction. The four-parameter Weibull curve associated with the profile of perturbation response estimated by the Levenberg–Marquardt method was used as input data for the models. Straightforward network topologies 4:3:1 (13 weights) and 4:2:1 (9 weights) for pyrogallol and gallic acid, respectively, allowed the analytes to be quantified with great accuracy (mean standard error of prediction for the generalization test) and precision (standard deviation) of 2.45% and 0.21 for pyrogallol and 7.61% and 1.63 for gallic acid. The selected model can be easily implemented via software by using simple quantification equations, from which significant chemical remarks can be inferred. Finally, the product unit neural network modelling offered better results when compared with sigmoidal neural networks and response surface analysis.

Keywords: Memetic algorithms; Product unit neural networks; Multi-component mixtures; Pyrogallol; Gallic acid; Oscillating reactions

1. Introduction

Simultaneous determination of closely related species is of continuing analytical attention and methods developed for resolving them constitute an interesting area in chemometrics [1–3]. Multivariate calibration techniques present some advantages over other methods as they are faster, sample treatment is usually reduced and the removal of interference is not strictly required in many cases. Nowadays, artificial neural networks (ANNs) are a powerful non-parametric nonlinear modelling technique in the field of simultaneous determination of several species in a given sample. The ANNs are able to provide models even when the information and data are complex, noise contaminated, nonlinear and incomplete as well as in the presence of any other unknown non-linearity in systems such as the analyte–analyte interaction (synergistic effects) [4–6]. Thus, ANNs have been shown to be a powerful tool for the resolution of multi-component samples in a great variety of areas of interest, such as in spectroscopic [7–10], electrochemical [11–13], kinetic [14–17] and chromatographic analysis, and in the case of the latter, especially for the quantification of unresolved peaks [18–22].

Multilayer perceptron (MLP) modelling based on different back-propagation algorithms has successfully been used for multi-component analysis. A recent trend in this and other fields of research is the development of ANN models with as limited number of weights as possible, which constitutes an additional significant objective to that of minimizing the errors in the generalization set. In fact, this kind of network model will be more appropriate to avoid over training, which increases its generalization ability over a new set of data. Several alternatives have been proposed to achieve this purpose. One is to select the suitable

* Corresponding author. Tel.: +34 957 212099.
E-mail addresses: ma1hemac@uco.es (C. Hervás), qa1sirom@uco.es (M. Silva).
[1] Tel.: +34 957 218349.

0169-7439/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.chemolab.2006.06.020

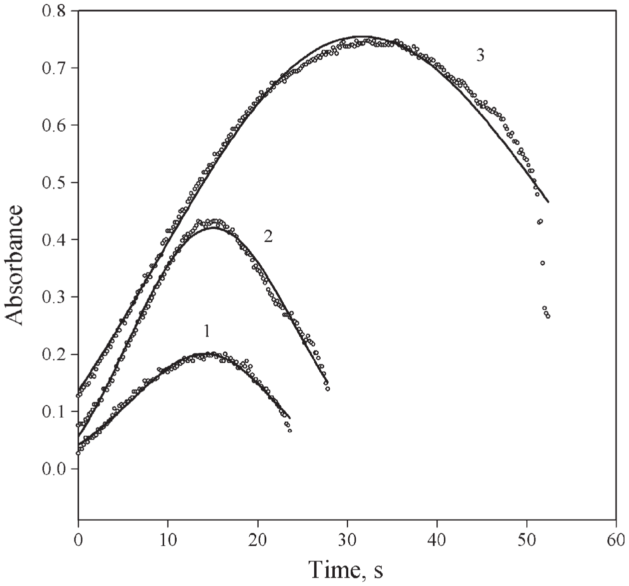


Fig. 1. Perturbation chemical oscillation responses fitted to a four-parameter Weibull function. (o) Experimental data and (–) Weibull curve. The concentrations of *P* and *GA* in the mixtures are as follow: (1) both at 1 μmol; (2) 1 and 4 μmol and (3) 4 and 1 μmol for *P* and *GA*, respectively.

input transformation variables from all the input variables available, for which principal component analysis is frequently used [7,14,15]. However, an interesting and powerful choice without losing some input information has been formerly proposed, which uses the estimated parameters of a predetermined function associated with the shape of the analytical signal as input data [19,20,23]. Thus, the estimates in the form of a three-parameter Gaussian [23] or a four-parameter Weibull curve [19,20] from kinetic [23] and chromatographic [19,20] data have been used as input neurons. The resulting network topologies allow the analytes in the mixture to be quantified with great accuracy and precision, and also with a great interpretability from a chemical point of view on account of their reduced dimensions. Another is based on the use of product unit neural networks (PUNNs) as a valuable alternative to MLP modelling. Product units enable a neural network to form higher-order combinations of inputs, having the advantages of increased information capacity and smaller network architectures when an interaction between the input variables occurs [24–27]. This kind of neural network model has proven to be a powerful tool for extracting useful analytical information in the quantification of analytes providing highly overlapping chromatographic peaks [19].

In this work, both alternatives are used for resolving multicomponent systems based on dynamic responses. For reducing the input information, the signal-time pair data are fitted to a four-parameter Weibull function by using the Levenberg–Marquardt (L–M) algorithm, and the resulting parameters are used as inputs for PUNNs. Different hybrid genetic algorithms (GA) based on the combination of an evolutionary algorithm, a clustering process and a descendent gradient local search procedure, are used for architectural design and estimation of weights. This synergy among diverse optimisation methods can provide different families of hybrid algorithms where the search is based on an explored first step followed by an exploited second one. The benefits of mutual interactions between a local and a global optimisation method working together have been studied in computational science, giving rise to techniques which have been called memetic algorithms (MAs), hybrid GAs, Lamarckian GAs, Balwinian GAs, etc. [28–32]. Recently, hybrid GAs with local search have been developed for optimisation in chemometrics studies involving discrete and continuous variables [33,34]. For continuous variables, the genetic algorithm uses a discrete codification system, in which the parameters are transformed in binary numbers by defining the lower and upper values that each parameter can adopt. In addition, a hybrid Lamarckian strategy is used for local optimisation and the fitness values obtained are applied to each individual in the population. Nevertheless, in this work we used a real codification in the evolutionary algorithm in such a way that parametric mutation operators are specifically designed for this type of codification, and the local search is not applied to all the individuals of the population due to the high computational cost that it represents. Thus, the local optimisation procedure is carried out in the last generation of the evolutionary algorithm because crossover operator is not used, and therefore, the recombination of the best individuals of one generation with those of the next is not necessary. The proposed methodology was validated with the simultaneous determination of pyrogallol (*P*) and gallic acid (*GA*), two closely related phenol derivatives, in mixtures due to their perturbing effects on the classical Belousov–Zhabotinskyi (BZ) by using the analyte pulse perturbation technique [35,36].

2. Theoretical background

The aim of the proposed approach is the automatic optimization of the structure and weights of PUNNs used for the simultaneous determination of closely related species based on the analysis of dynamic signals; concretely, those responses yielded by the perturbation of a mixture of analytes to oscillating chemical reactions [35]. The first step of this approach consists of extracting the information from the analytical response in order to select the inputs to the PUNNs models. Upon examining this response, it can be observed that the signals set (t_i,S_{t_i}) can be accurately fitted by least-square regression to a four-parameter Weibull curve, defined by S_m (maximum signal), *B* (dispersion of the analytical signal values from S_m), *C* (related to the function shape, which is associated to the inflection points of the curve and defines the concavity

Table 1
Parameter value±standard deviation obtained for the fitting of the four-parameter Weibull function to typical perturbation oscillating responses provided by pyrogallol/gallic acid mixtures

Parameter	Assayed mixtures, [*P*]:[*GA*]		
	1:1	1:4	4:1
S_m	0.210±0.001	0.421±0.002	0.756±0.003
B	27.3±1.4	22.7±0.3	49.6±0.7
C	3.74±0.21	2.72±0.05	2.53±0.06
t_m	14.12±0.09	15.04±0.06	31.64±0.14
R^2	0.9818	0.9909	0.9782

and convexity regions), and t_m (time corresponding to the maximum). Fig. 1 shows the fit provided by the four-parameter Weibull function on typical perturbation BZ reaction responses obtained for three mixtures of the assayed phenol derivatives. Upon examining the curves and from the estimated statistical parameters (see Table 1), it can be concluded that the Weibull function is a fine tool for modelling this kind of dynamic responses.

In the second step, a procedure is used to construct models based on the learning of the neural networks from the patterns of the training set by using a recently reported hybrid GAs [37]. In the PUNN models proposed in this work, the potential base function is as follows:

$$B_j(x, w_j) = \prod_{i=1}^{m} x_i^{w_{ji}} \tag{1}$$

where $\boldsymbol{w}_j=(w_{j1}, w_{j2}, \ldots, w_{jm})$ is a set of coefficients for the potential base functions. The proposed model is a lineal combination of the p base functions:

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j \left(\prod_{i=1}^{m} x_i^{w_{ji}} \right) \tag{2}$$

where $x \in S \subset R^m$ and S is a sub-set, with $x > 0$, of the space of dimension m, $\beta=(\beta_0, \beta_1, \ldots \beta_P)$ the vector of coefficients multiplying each of the base functions of the set $\boldsymbol{B}_{\mathrm{F}}=\{1, \boldsymbol{B}_1(x, w_1), \ldots, \boldsymbol{B}_p(x, w_p)\}$, and w_j the coefficients associated with the base functions, which introduce non-linearity into the model.

For the data set $D=\{x_l, y_l\}$, for $l=1, \ldots, n$, the regression model can be expressed by means of a potential base function topology, $f{:}A \subset R^m$ such that

$$Y_l = f(x_1) = \beta_0 + \sum_{j=1}^{p} \beta_j \left(\prod_{i=1}^{m} x_{li}^{w_{ji}} \right) + e_l \qquad l = 1, 2, \ldots, n \tag{3}$$

where β_j and $w_{ji} \in R$, and p, $m \in N$.

If the habitual methods of estimation of parameters of a linear model are followed, in this case the procedure is not trivial because the design matrix $\mathbf{B}$

$$\mathbf{B} = \begin{pmatrix} \mathbf{B_0}(\mathbf{x_1}, \mathbf{w_1}) & \ldots & \mathbf{B_p}(\mathbf{x_1}, \mathbf{w_p}) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \mathbf{B_0}(\mathbf{x_n}, \mathbf{w_1}) & \ldots & \mathbf{B_p}(\mathbf{x_n}, \mathbf{w_p}) \end{pmatrix} \tag{4}$$

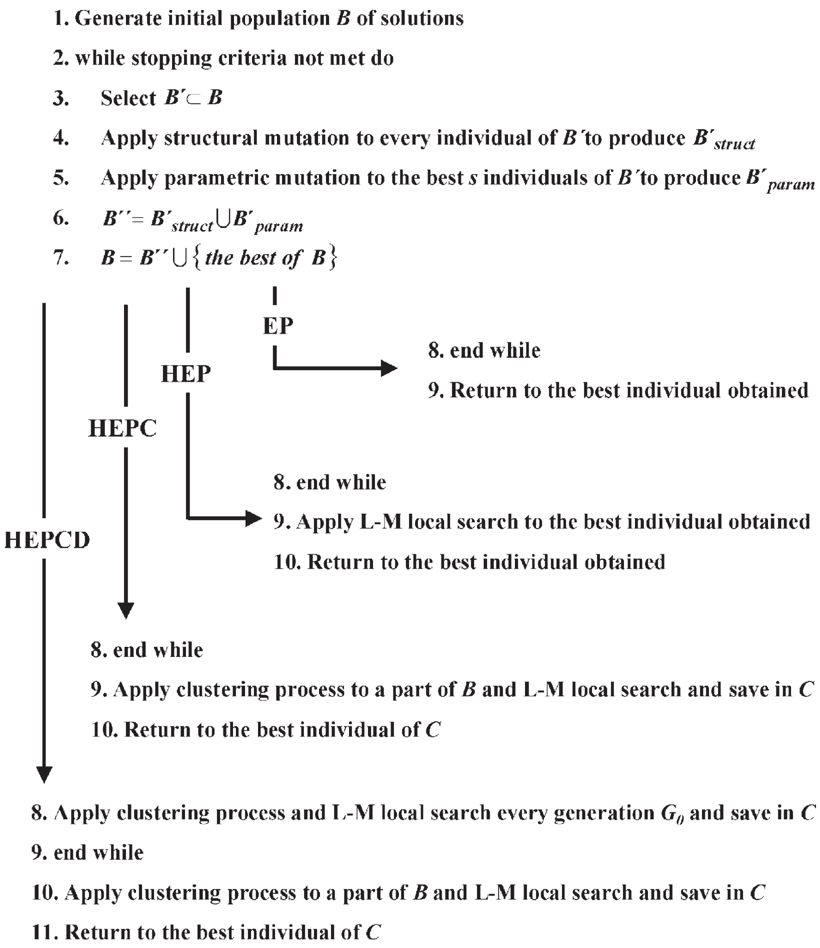
depends in turn on the w_j parameters. In this way, the elements of the $\mathbf{B}$ matrix are potential functions of the x values. Thus, a hybrid GAs is used to estimate the optimum values of p and those of the β_j and w_j coefficients. This kind of functions can be represented by a neural network, which, in this work, has the following architecture: one input layer consisted of the estimated four-parameter Weibull function; one hidden layer with an appropriate number of nodes, and one output layer providing the concentration of the analytes to be determined in the assayed mixtures: [P] and [GA].

2.1. Evolutionary algorithms

GAs are a class of optimization algorithms, based in a population of solutions, which are efficient for exploring the entire search space. They are, however, relatively poor at finding the precise local optimal solution in the region into which the algorithm converges. Many researchers have shown that GAs perform well for global searching because they are capable of quickly finding and exploiting promising regions of the search space, but they take a relatively long time to converge to a local optimum [38,39]. In the past decade, new approaches have been reported for improving GAs using local-improver procedures (LIPs), which search a “neighbourhood” of the starting solution until either the first improvement or the best improvement (local optimum) is found. These new methodologies are based on the combination of LIPs, which are good at finding local optima (local exploiter), and evolutionary algorithms (global explorer). One of these methods is the so-called memetic algorithms [40–43], a kind of search strategy in which a population of optimizing agents synergistically cooperates and competes, and where, unlike traditional evolutionary computation methods, all available knowledge about the problem under study can be exploited. In this paper we used a local search Quasi-Newtonian algorithm, the L–M one, because it is especially suitable for regression problems where the sum-of-squares residual should be minimized. However, the particular combination of GA with local search is extremely important in terms of possible solution quality and computational efficiency; and therefore the right mixture of local exploitation *versus* global exploration should be found.

The high-level template of an MA algorithm is a generation with three main components: selection, recombination, and replacement. In the proposed algorithms the recombination is carried out using two types of mutation operators: parametric and structural mutation. These operators generate a new solution by partly modifying an existing solution. Thus, in parametric mutation, the values of the coefficients of the model are changed adding to a gene chosen at random of the chromosome a Gaussian variable; whereas in the structural mutation, the new solution is obtained adding or removing addends in the model of network (nodes); adding, removing and/or re-defining coefficients of the model (connections).

In this paper, different MA methodologies are proposed for obtaining the structure and weights of neural networks based on product units used for resolving multi-component mixtures. The methodology is based on the combination of an evolutionary programming algorithm (global explorer), a clustering process (semi-global explorer) and a local improvement procedure carried out by the L–M algorithm (local exploiter). Different versions of a MA are used depending on the stage at which the local search and the cluster partitioning are carried out: (1) an evolutionary algorithm without the clustering process and local search is applied, which is represented by (EP); crossover is not used as this operation is usually regarded as less effective for network evolution [44]; (2) a hybrid evolutionary programming (HEP) is tested; the EP is run without the local optimization algorithm and is then applied to the final best solution obtained by the EP. This allows the precise local optimum around the final



```
1. Generate initial population $B$ of solutions
2. while stopping criteria not met do
3.     Select $B' \subset B$
4.     Apply structural mutation to every individual of $B'$ to produce $B'_{struct}$
5.     Apply parametric mutation to the best $s$ individuals of $B'$ to produce $B'_{param}$
6.     $B'' = B'_{struct} \cup B'_{param}$
7.     $B = B'' \cup \{the\ best\ of\ B\}$

EP
    8. end while
    9. Return to the best individual obtained

HEP
    8. end while
    9. Apply L-M local search to the best individual obtained
    10. Return to the best individual obtained

HEPC
    8. end while
    9. Apply clustering process to a part of $B$ and L-M local search and save in $C$
    10. Return to the best individual of $C$

HEPCD
    8. Apply clustering process and L-M local search every generation $G_0$ and save in $C$
    9. end while
    10. Apply clustering process to a part of $B$ and L-M local search and save in $C$
    11. Return to the best individual of $C$
```

Fig. 2. Pseudo codes of the evolutionary programming algorithms.

solution to be found; (3) the HEP with the clustering algorithm (HEPC) is used; the clustering process is applied in each model's space of fitness for a large enough subset of the best individuals of the final population. In this method it is very important to determine the rate of best individuals to consider, as well as the number of clusters. After that, the L–M algorithm is applied to the best individual in each cluster; and (4) the so-called dynamic hybrid evolutionary programming with clustering (HEPCD) carries out both the clustering process and the L–M local search dynamically every G_t generation, where G_t must be fixed by the user. The final solution is the best individual among the local optima found during the evolutionary process. To clarify these methodologies, they are briefly described below.

2.2. Evolutionary programming algorithm

The EP algorithm used in this work for architecture design and estimation of real-parameters has common points with other evolutionary algorithms reported in the literature [44–46]. It begins the search with an initial population, which is updated in each iteration using a population-update algorithm. The population is subject to the operations of replication and mutation.

In the EP algorithm for the learning of the neural network, the number of nodes in the hidden layer is taken from a uniform distribution in the interval (0, m/2], where m is the maximum number of hidden nodes in any network of the population. In this way, the initial population is formed from models easier than the most complex possible one. The number of connections between each node of the hidden layer and the input nodes is determined from a uniform distribution in the interval $[0, k]$, k being the number of independent variables. At least, there is one connection between the hidden layer and the output node. Once the topology of the network is defined, a weight is assigned to each connection from a uniform distribution in the interval $[-L, L]$ for weights between the input and hidden layers, and in the interval $[-M, M]$ for the weights between the hidden layer and the output node. In this form, we obtained the initial population that constitutes the base solution B. The pseudo code for the EP algorithm is shown in Fig. 2.

2.3. Fitness assignment and parametric and structural mutation

Let $D=\{(x_l,y_l)\colon l=1, 2,\ldots, n\}$ be the training data set. We consider the square error of an individual $f(x,\theta)$ of the population:

$$J(\theta) = \frac{1}{2}\sum_{l=1}^{n}\left(\mathrm{y}_l - f(x_l,\theta)\right)^2 \tag{5}$$

and we define the fitness function $A(f(x,\theta))$ as the increasing transformation of the error following:

$$A(f(x,\theta)) = \frac{1}{1+J(\theta)} \tag{6}$$

For parametric mutation we use a simulated annealing algorithm [47,48] and the severity of a mutation to an individual $f(x,\theta)$ is dictated by the temperature, $T(f(x,\theta))$, given by:

$$T(f(x,\theta)) = 1{-}A(f(x,\theta)), \quad 0 \leq T(f(x,\theta)) < 1 \tag{7}$$

Parametric mutation is accomplished for each coefficient w_{ji}, β_j of the model in Eq. (3) with Gaussian noise, where the variance of the normal distribution depends on the temperature:

$$w_{ji}(t+1) = w_{ji}(t) + \xi_1(t) \text{ and } \beta_j(t+1) = \beta_j(t) + \xi_2(t) \tag{8}$$

where $\xi_k(t) \in N(0, \alpha_k(t)T(f(x,\theta)))$, $k=1$, 2 represents a one-dimensional normally distributed random variable with mean 0 and variance $\alpha_k(t)T(f(x,\theta))$. The parameters $\alpha_k(t)$, $k=1$, 2, allowing the adaptation of the learning process and change along the evolution:

$$\alpha_k(t+1) = \begin{cases} (1+\beta)\alpha_k(t) & \text{if } A(g) > A(g{-}1), \forall g \in \{t, t{-}1, \ldots, t{-}\rho\} \\ (1{-}\beta)\alpha_k(t), & \text{if } A(g) = A(g{-}1), \forall g \in \{t, t{-}1, \ldots, t{-}\rho\} \\ \alpha_k(t) & \text{otherwise} \end{cases} \tag{9}$$

$k=1$, 2, where $A(g)$ is the fitness of the best individual in generation t. It should be pointed that the modification of the exponents w_{ji} is different from the modification of the coefficients β_j, $\alpha_1(t) \ll \alpha_2(t), \forall t \in R$.

Structural mutation implies a modification of the structure of the function and allows the explorations of different regions of the search space and helps to keep the diversity of the population. There are five different structural mutations: Node addition (AN), node deletion (EN), connection addition (AC), connection deletion (DC) and node fusion (UN). The four first ones are similar to the mutations of GNARL model [44]. In the UN, two randomly selected nodes, a and b, are replaced by a node c, which is a combination of both. The connections that are common to both nodes are kept, with a weight given by:

$$\beta_c = \beta_a + \beta_b, \quad w_{ic} = \frac{w_{ia} + w_{ib}}{2} \tag{10}$$

The connections that are not shared by the nodes are inherited by c with probability of 0.5 and its weight is unchanged. For each mutation (excepting node fusion) there is a minimum value, Δ_{MIN}, and a maximum value, Δ_{MAX}, and the number of elements (nodes and connections) involved in the mutation is calculated as:

$$\Delta_{MIN} + \lfloor uT(f(x,\theta))(\Delta_{MAX}{-}\Delta_{MIN}) \rfloor \tag{11}$$

where u is a random uniform variable in the interval [0, 1]. All the above mutations are made sequentially in the given order, with probability $T(f(x,\theta))$, in the same generation on the same network. If the probability does not select any mutation, one of the mutations is chosen at random and applied to the network.

2.4. Hybrid algorithms

Regarding the hybrid algorithms, in the HEP the L–M algorithm is applied to the best individual obtained by the EA in the final generation, whereas in the HEPC, the clustering process is applied to the best $\tilde{s}N_p$ individuals of the final generation, which is divided into K clusters C_1, C_2,…, C_k using a standard K-means algorithm. After that, the L–M algorithm is applied to the best individual of each cluster. The individuals obtained with the local-search in each cluster are included in a set C (local optimum set) (see Fig. 2).

In the HEPCD, the clustering process and the L–M algorithm are applied to the best individual of each cluster every G_0 generation and in the final one. The clustering process is applied to the best $\tilde{s} \times N_p$ individuals of the current population. The individuals obtained with the local-search in each cluster are included (stored) in a set C (dynamic local optimum set). The final solution is the best individual among the local optimums of set C (see Fig. 2).

3. Experimental

Twenty-seven perturbation responses provided by samples containing uniformly distributed concentrations of pyrogallol (1.0–6.0 μmol) and gallic acid (1.0–7.0 μmol) were prepared in triplicate as described elsewhere [23]. Three experimental designs consisting of an interpolation between different concentrations of each phenol derivative were used for selecting the training and generalization sets. As can be seen in Fig. 3, the three data sets tested were as follows: A (train/test 21:6), B (train/test 18/9) and C (train/test 15/12), although it should be considered that each synthetic mixture was prepared in triplicate.

The L–M algorithm was used to obtain the four estimated coefficients of the Weibull function. Convergence of the iterative

Table 2
Values of the parameters used by the algorithms

Parameter	Value/Interval
Evolutionary algorithm	
N_R	1000
m	8
[−M,M]	[−5, 5]
[−L,L]	[−5, 5]
s	0.1
Parametric mutation	
$\alpha_1(0)$	1
$\alpha_2(0)$	5
β	0.1
ρ	10
Structural mutation	
Node addition	[1, 2]
Node deletion	[1, 3]
Connection addition	[1, 2m]
Connection deletion	[1, 3m]
Hybrid algorithms	
r	90
K	6
$\tilde{s}$	0.25
G_0	800

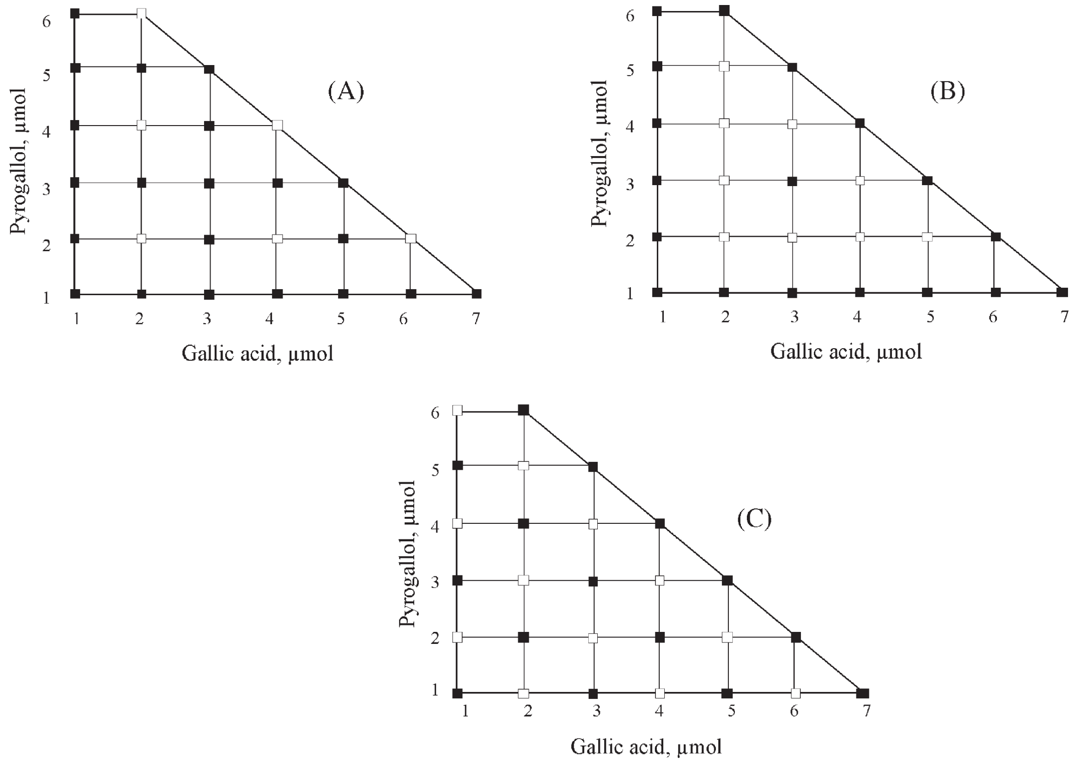


Fig. 3. Experimental designs assayed for selecting the training (■) and the generalization (□) sets for optimizing ANN architectures.

process was achieved with a tolerance of 0.0001 and a maximum number of iterations of 100. To homogenize the values of the input nodes, they were scaled over the range 0.1 and 0.9. Thus, the new-scaled variables were expressed as follows:$\hat{S}_m^*$, $\hat{t}_m^*$, $\hat{B}^*$ and $\hat{C}^*$. In a similar way, the concentration of the analytes in the sample, defined by [*P*] and [*GA*], were used as outputs for the tested network models, and also scaled, in this case over the range 1 and 2. These new-scaled independent variables were designated as [$\hat{P}^*$] and [$G\hat{A}^*$]. After optimizing the network models, estimations should be de-scaled according to the same equation.

The MAs for ANN models, in C language, were run on a PC Pentium IV compatible computer. The principal parameters used in the algorithms are shown in Table 2. The stop criterion was reached whenever one of the following two conditions was fulfilled: i) The algorithm achieved a given fitness; and ii) for 20 generations there was no improvement either in the average performance of the best 20% of the population or in the fitness of the best individual. An analysis of variance was used for adjusting these meta-parameters. The analysis had also shown that the algorithm was quite robust to the modification of the values of the parameters within reasonable ranges. For the determination of [*P*] the initial architecture was 4:3:1 with bias and the maximum number of generations was 3000, and for [*GA*] the initial architecture was 4:2:1 with bias and the maximum number of generations was 2400.

The performance of the algorithms was tested using various network topologies that were run thirty times. The accuracy of each model was assessed in terms of the standard error of prediction (SEP) for the results obtained for both data sets, that

Table 3
Accuracy and statistical results for SEP_T and SEP_G provided by the algorithms used for the resolution of mixtures of pyrogallol and gallic acid based on the assayed experimental sampling designs

Algorithm	Design A (21/6)				Design B (18/9)				Design C (15/12)			
	SEP_T		SEP_G		SEP_T		SEP_G		SEP_T		SEP_G	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
Pyrogallol												
EP	167.6	5.9	216.3	73.7	3.36	0.80	3.54	1.07	3.11	0.44	4.58	0.74
HEP	40.1	22.5	4.1e3	2.4e2	2.21	1.06	2.92	0.82	1.98	0.42	3.11	0.64
HEPC	35.3	12.7	1.2e4	2.3e4	1.82	0.96	2.75	0.55	1.66	0.37	2.86	0.57
HEPCD	14.9	10.5	4.3e3	3.7e3	1.26	0.23	2.45	0.21	1.43	0.17	2.51	0.30
Gallic acid												
EP	10.3	0.99	9.38	2.86	10.0	1.61	8.28	2.82	7.98	1.66	12.4	2.17
HEP	9.61	1.17	9.58	2.63	9.18	1.79	8.34	2.98	7.40	1.66	11.8	2.08
HEPC	9.21	1.33	10.3	2.74	8.76	1.38	8.06	1.70	6.96	0.83	11.0	1.96
HEPCD	8.63	0.99	7.80	1.62	8.53	1.22	7.61	1.63	6.65	0.68	10.5	1.63

Table 4
Accuracy and statistical results provided by the four evolutionary algorithms used for designing PUNN models (s=25%, K=4 and G=4 for [P] and 3 for [GA])

Algorithm	SEP_T				SEP_G				# conn		# eval	
	Mean	S.D.	Best	Worst	Mean	S.D.	Best	Worst	Mean	S.D.	Mean	S.D.
Pyrogallol												
EP	3.36	0.80	2.04	4.86	3.54	1.07	1.99	6.26	11.8	1.14	1060466	385403
HEP	2.21	1.06	1.17	4.77	2.92	0.82	1.80	5.66	11.8	1.14	1060494	385407
HEPC	1.81	0.96	0.97	4.55	2.74	0.55	1.79	4.66	12.0	1.35	1060642	385443
HEPCD	1.26	0.23	0.97	2.11	2.45	0.21	1.80	2.87	11.7	1.42	1060994	394421
Gallic acid												
EP	10.0	1.61	7.64	15.6	8.28	2.82	4.86	21.9	9.5	0.94	1337861	142551
HEP	9.18	1.79	7.28	15.3	8.34	2.98	5.63	21.5	9.5	0.94	1337906	142564
HEPC	8.76	1.38	7.27	11.8	8.06	1.70	5.62	12.0	9.5	0.82	1338012	142584
HEPCD	8.53	1.22	7.27	11.8	7.61	1.63	5.62	11.9	9.4	0.96	1338314	150939

is, SEP_T for the training set, and SEP_G for the generalization set, whereas the robustness was assessed in terms of standard deviation (S.D.). In this way, the SEP was calculated as:

$$\mathrm{SEP}=\frac{100}{\bar{A}_i}\sqrt{\frac{\sum_{i=1}^{n}(A_i-\hat{A}_i)^2}{n}};\ \mathrm{SEP}=\frac{100}{\bar{A}_i}\sqrt{\mathrm{MSE}} \qquad (12)$$

where A_i and $\hat{A}_i$ are the experimental and expected values for the analyte concentration in the mixture, $\bar{A}_i$ is the mean of the experimental values of the training set, or of the generalization set, and n is the number of patterns used (n_T for the training set and n_G for the generalization set). We used the SEP instead of the mean standard error (MSE) because it is an adimensional measure that allows comparing results of error for dependent variables with different scale. The parametric Student's t and the non-parametric Kolmogorov–Smirnov (K–S) were performed using SPSS 12.0 statistical software [49], and used to evaluate the performance of the different models in selecting the most suitable network architecture.

4. Results and discussion

The aim of this work was to develop a methodology for the resolution of mixtures of closely related species by using ANNs designed and learned by MAs. The approach was tested on the simultaneous determination of two related phenol derivatives, namely P and GA based on their perturbation effect on the classical Belousov–Zhobotinskyi oscillating chemical reaction [23,36]. The analytical response obtained when this nonlinear system was perturbed by injecting a micro-volume of the mixture of both phenols was the combination of the signals provided by the following chemical processes [36]: (a) the oscillating reaction; (b) the oxidation of phenol derivatives; and (c) the presence of synergistic effects, which cause strong interactions among the input variables of the ANN models, and therefore PUNN models are recommended. In these conditions, the profile of the response curve changed as a compromise of these contributions (see Fig. 1).

One important issue to be addressed for obtaining the best generalization capability of the assayed ANN models is the composition of the data set used for the training and generalization tests. As can be seen in Fig. 3, three different data sets were tested for this purpose from all the data (81 samples) obtained by preparing 27 synthetic samples in triplicate as described under Experimental. These designs are based on the combination of the so-called grid sampling and judgmental one [50]. In fact, in grid sampling, samples are taken at regularly spaced intervals over the concentration of both components, and in the judgmental sampling the selection of sampling units is based on knowledge of the feature under research and on professional judgment. Thus, the design A evaluates the extrapolation ability of the PUNN models in three mixtures of the generalization set; the design B is a priori the most robust because the training set covers samples with higher and lower concentrations of both phenols in mixtures; and the design C is an intermediate situation between the former ones. To evaluate these designs, one output PUNN model was used, considering the output layer as a single node corresponding to the concentration of the phenol in the mixture to be analyzed. Table 3 shows the statistical results obtained over 10 runs using PUNN models optimized by the four evolutionary algorithms tested. As can be seen, the design B (18/9) provides more accurate and robust results (smaller SPE_G and S.D. values) for the resolution of mixtures of both phenols. Regarding the extrapolation ability of the PUNN models [see results from the design A (21/6)], the results for gallic acid are similar to those found using the other designs; however, erratic results were obtained for pyrogallol, which can be ascribed to the high concentration of this phenol in three samples including in the generalization set (see Fig. 3), which provoked a strong distortion of the analytical response profile as can be seen in Fig. 1.

Table 5
Significant levels (p-values) for the correlation and t-tests for SEP_G

Test	Pyrogallol		Gallic acid	
	Corr-test	t–test	Corr-test	t–test
$\mu_{EP}-\mu_{HEP}=0$	0.000	0.000 (*)	0.000	0.862
$\mu_{EP}-\mu_{HEPC}=0$	0.009	0.000 (*)	0.282	0.695
$\mu_{EP}-\mu_{HEPCD}=0$	0.196	0.000 (*)	0.160	0.209
$\mu_{HEP}-\mu_{HEPC}=0$	0.000	0.077 (***)	0.000	0.516
$\mu_{HEP}-\mu_{HEPCD}=0$	0.009	0.002 (*)	0.009	0.082 (***)
$\mu_{HEPC}-\mu_{HEPCD}=0$	0.001	0.002 (*)	0.001	0.016 (**)

4.1. Selection of memetic algorithm used for optimization PUNN models

To compare the predictive ability of PUNN models optimized by the four evolutionary algorithms in terms of SEP_T, SEP_G, robustness (S.D.), number of connections (# conn) and number of evaluations (# eval), neural network models with a single node in the output layer (concentration of phenol in the mixture to be determined) were made, and the experimental sampling design B (18/9) was used for establishing the training and generalization sets. The 4:3:1 and 4:2:1 architectures were chosen to start the model selection processes for *P* and *GA*, respectively, both over 30 runs.

As can be seen in Table 4, all models provided quite good results (in terms of accuracy and precision) for determining the concentration of each phenol in the assayed mixtures. The HEPCD algorithm, however, yielded better results: smaller SEP mean values, SD and # conn, although # eval was slightly greater. For *P*, the proposed PUNN method is more robust, taking into account the lower SD values provided for both SPE_T and SPE_G.

Table 5 shows the *p*-values of two-tailed t-tests for dependent distributions (see *p*-values of a standard test of dependence), comparing the means (over 30 runs) of the algorithms in pairs. Previously, a standard K–S normality test showed that the results for all the methodologies were normal. For *P*, the *t*-tests show that HEPCD is, in mean, the best performing algorithm, at a confidence level of 1% (denoted by * in the table) when compared with the other algorithms, and that HEPC is better than HEP and EP at a confidence level of 10 (denoted by *** in the table) and 1%, respectively. For *GA*, there only exist significant differences, in mean, within HEPCD and HEPC at 5% (denoted by ** in the table), and HEPCD with HEP for 10% confidence levels. With respect to the size of the network, or the number of connections, there is not a significant difference, in mean, for the EP versus HEPCD methodologies, for both *P* and *GA* at a confidence level of 5%.

According to this study, the HEPCD methodology was selected because it offers greater accuracy and precision (smaller SPE_G mean and S.D. values) although resulting in a slight increase in network complexity. From this PUNN model, the following quantitative equations can be drawn for the simultaneous determination of each phenol in mixtures:

$$\begin{aligned}[\widehat{P}]^* &= 0.810 + \widehat{P}U_1 + \widehat{P}U_2 + \widehat{P}U_3\\ [\widehat{P}]^* &= 0.810 + 1.261(\widehat{S}^*_m)^{4.583}(\widehat{B}^*)^{1.819} + 0.778(\widehat{S}^*_m)^{0.083}\\ &\quad \times (\widehat{C}^*)^{-0.103}(\widehat{t}^*_m)^{0.605} - 10.134(\widehat{S}^*_m)^{6.346}(\widehat{B}^*)^{3.329}\\ &\quad \times (\widehat{C}^*)^{1.110}(\widehat{t}^*_m)^{1.822}\end{aligned} \tag{13}$$

$$\begin{aligned}[\widehat{GA}]^* &= 0.964 + \widehat{P}U_1 + \widehat{P}U_2\\ [\widehat{GA}]^* &= 0.964 + 1.243(\widehat{S}^*_m)^{1.142}(\widehat{G}^*)^{0.173}(\widehat{t}^*_m)^{-0.416} -\\ &\quad -0.189(\widehat{B}^*)^{0.895}(\widehat{C}^*)^{-0.590}(\widehat{t}^*_m)^{-0.306}\end{aligned} \tag{14}$$

which are based on the parameters estimated by the Weibull regression of the analytical response, the optimized network weights and the product unit transfer functions.

Finally, in order to evaluate the potential of the proposed PUUN model, a comparative study was performed involving two common methodologies such as MLPs (sigmoidal neural networks) [23] and response surface analysis (RSA). Although the resulting conclusions will be supported in the results provided by the assayed chemical system, they can also be extended to other ones providing similar analytical responses, such as chromatographic peaks, spectrometric spectra, etc. As can be seen in Table 6, the use of PUNN models can be a useful choice with respect to the alternatives tested, because it provides simpler and more robust models with smaller SEP_G values. The behaviour can be assigned to the greater ability of the PUNNs for modelling systems with stronger interaction between the input variables, as in the simultaneous determination of *P* and *GA* mixtures, which involve synergistic effects.

4.2. PUNN models designed by the HEPCD: Chemical interpretation

As stated above, the use of MAs for designing network models enables architectures with reduced dimensions to be obtained in such a way that the mathematical transformation between the input and output can be easily implemented (see Eqs. (13) and (14)). From these simpler models, quality chemical information could be derived to explain the analytical problem at hand. For this purpose, it is necessary to distinguish how the relative concentration of both phenols in the mixtures affects the profile of the analytical response (see Fig. 1). As can be seen, the two phenols exhibit quite different kinetic behaviour. Thus, mixtures with the same concentration of *P* and variable concentrations of *GA* (curves 1 and 2) reveal an increase in the maximum, S_m, whereas the time corresponding to it, t_m, remains virtually constant. Different behaviour was observed in mixtures with a constant concentration of *GA* and variable concentrations of *P* (curves 1 and 3): both S_m and t_m increase with the increasing in the [*P*], although the most significant observation is the strong change in the profile of the response curve as the concentration of *P* increases. This performance is closely related to the values of the *B* and *C* parameters provided by the Weibull function used for fitting the analytical data. In light of these considerations and using the

Table 6
Comparison of the quality achieved in the resolution of pyrogallol/gallic acid mixtures by using PUNN, MLP and SRA

Model	Pyrogallol		Gallic acid	
	SEP_G	# Weights	SEP_G	# Weights
PUNN	1.80	13	5.62	9
MLP [a]	4.01	14	8.98	15
SRA [b]	4.43	10	35.26	11

[a] From Ref. [23].

[b] The equations derived for each model are as follows:

$$\begin{aligned}[\widehat{P}] &= -3.853 - 0.101\widehat{B} + 0.643\widehat{C} + 0.514\widehat{t}_m - 0.577\widehat{S}_m\widehat{B}\\ &\quad +0.648\widehat{S}_m\widehat{t}_m - 0.017\widehat{B}\widehat{t}_m - 0.042\widehat{C}\widehat{t}_m + 6.709\widehat{S}^2_m + 0.011\widehat{B}^2\,(R^2 = 0.9989)\\ [\widehat{GA}] &= 19.707 + 18.430\widehat{S}_m + 0.590\widehat{B} - 3.744\widehat{C} - 2.539\widehat{t}_m\\ &\quad +0.882\widehat{S}_m\widehat{B} - -1.877\widehat{S}_m\widehat{t}_m + 0.061\widehat{B}\widehat{C} + 0.077\widehat{B}\,\widehat{t}_m\\ &\quad +0.089\widehat{C}\widehat{t}_m - 0.038\widehat{B}^2\,(R^2 = 0.9875)\end{aligned}$$

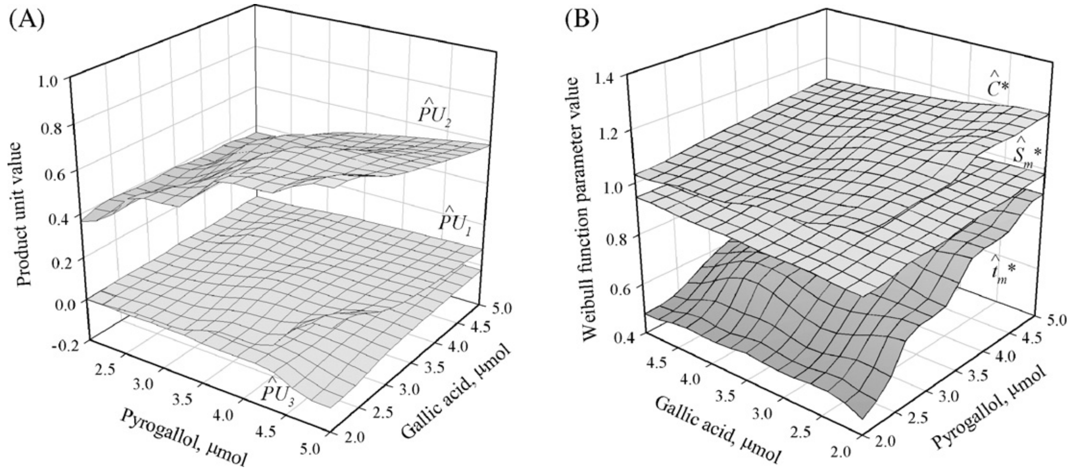


Fig. 4. Relative contributions of (A) the product unit terms, and (B) the Weibull function parameters used for the quantitative determination of the pyrogallol by using a PUNN model designed by the HEPCD algorithm.

Eqs. (13) and (14), the most notable features of the proposed model for each phenol are discussed below from both computational and chemical points of views.

Model for pyrogallol. In order to evaluate the relative influence of each product unit (PU) of Eq. (13), a three-dimensional plot has been made of each *PU* term as a function of the *P* and *GA* concentrations in the mixtures subjected to analysis in the generalization set (Fig. 4A). As can be seen, the PU_2 term can be considered as the basic one for determining the concentration of *P* in the assayed mixtures. The other terms show a slight effect for mixtures with a high concentration of *P*. In order to consider the relative effect of the Weibull function parameters involved in PU_2, another three-dimensional plot was made, but in this case the *Z*-axis represents the value of each parameter with its

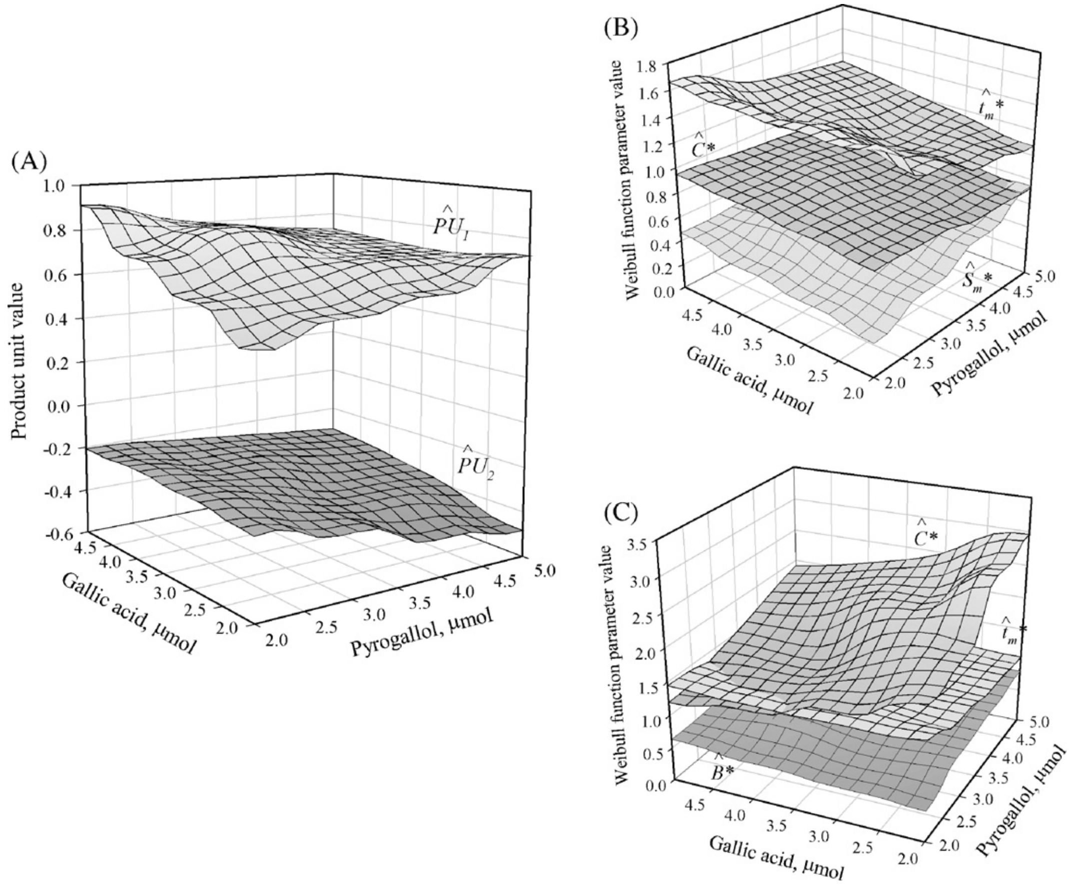


Fig. 5. Relative contribution of (A) the product unit terms, and (B) and (C) the Weibull function parameters used for the quantitative determination of gallic acid by using a PUNN model designed by the HEPCD algorithm.

corresponding exponent (Fig. 4B). From this figure it can be inferred that the value of $[\hat{P}]^*$ depends mainly on $\hat{t}_m^*$, the contribution of $\hat{S}_m^*$ and $\hat{C}^*$ being practically negligible, except for mixtures with high concentrations of *P*. In short, the $\hat{t}_m^*$ parameter is the key for the determination of *P* in the mixture. This conclusion is in agreement with the behaviour observed for this phenol in the response curves shown in Fig. 1.

Model for gallic Acid. Fig. 5 shows similar three-dimensional plots as stated above, but in this case, for *GA* supported on Eq. (14). As can be seen in Fig. 5A, both *PU* terms exhibited an appreciable influence on the determination of $[\hat{G}A]^*$, although the PU_1 term shows a slightly greater influence. From the results shown in Fig. 5B and 5C, it follows that the value of $[\hat{G}A]^*$ depends basically on the $\hat{S}_m^*$ and $\hat{C}^*$ according to PU_1 Fig. 5B), and essentially on $\hat{C}^*$ considering PU_2 (Fig. 5C). In both cases, $\hat{t}_m^*$ shows only a lesser influence for mixtures with smaller concentrations of *GA*, whereas the effect of $B\hat{}^*$ is practically negligible. In summary, the determination of $[\hat{G}A]^*$ in the mixture depends basically on $\hat{C}^*$ and to a lesser extent on $\hat{S}_m^*$ and $\hat{t}_m^*$ in this order of importance. In other words, the presence of infection points in the analytical response (associated to $\hat{C}^*$) is closely related to the concentration of *GA* in the mixture. Again, this computational conclusion is in agreement with experimental data as shown in Fig. 1 by comparing curves 1 and 2.

5. Conclusions

As shown in this study, dynamic hybrid evolutionary programming with clustering is a useful choice for trained and designed multiplicative neural networks based on product units, which has proven to be a powerful tool for the simultaneous determination of analytes in mixtures. So, the ensuing PUNN models provide good information ability, small network architectures, and robust models and they are quite simple and easier to interpret from a chemical point of view. Simple and clear relationships can be established between the input variables (the four parameters of the Weibull curve fitted to the dynamic response) and the output variable (the concentration of the analyte in the mixture). In addition, they show better performance when compared to sigmoidal neural networks and surface response analysis.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the Spanish Department of Research of the Ministry of Education and Science under the BQU2003-03027 (Department of Analytical Chemistry, University of Cordoba) and TIC2002-04036-C05-02 (Department of Computer Science, University of Cordoba) Projects. FEDER also provided additional funding.

References

[1] R.G. Brereton, Analyst 125 (2000) 2125–2154.
[2] M. Meloun, J. Militký, M. Hill, R.G. Brereton, Analyst 127 (2002) 433–450.
[3] R.N. Feudale, N.A. Woody, H.W. Tan, A.J. Myles, S.D. Brown, J. Ferre, Chemom. Intell. Lab. Syst. 64 (2002) 181–192.
[4] J. Zupan, J. Gasteiger, Anal. Chim. Acta 248 (1991) 1–30.
[5] J.R.M. Smits, W.J. Melssen, L.M.C. Buydens, G. Kateman, Chemom. Intell. Lab. Syst., Lab. Inf. Manag. 22 (1994) 165–189.
[6] F. Despagne, D.L. Massart, Analyst 123 (1998) 157R–178R.
[7] Y.N. Ni, G.W. Zhang, S. Kokot, Food Chem. 89 (2005) 465–473.
[8] N. Maleki, A. Safavi, F. Sedaghatpour, Talanta 64 (2004) 830–835.
[9] D. Mikami, T. Ohki, K. Yamaji, S. Ishihara, D. Citterio, M. Hagiwara, K. Suzuki, Anal. Chem. 76 (2004) 5726–5733.
[10] F. Dieterle, S. Busche, G. Gauglitz, Anal. Chim. Acta 490 (2003) 71–83.
[11] M. Shamsipur, J. Tashkhourian, B. Hemmateenejad, H. Sharghi, Talanta 64 (2004) 590–596.
[12] Y.N. Ni, P. Qiu, S. Kokot, Anal. Chim. Acta 516 (2004) 7–17.
[13] E. Richards, C. Bessant, S. Saini, Analyst 129 (2004) 355–358.
[14] Y.N. Ni, C.F. Huang, S. Kokot, Chemom. Intell. Lab. Syst., Lab. Inf. Manag. 71 (2004) 177–193.
[15] M. Blanco, J. Coello, H. Iturriaga, S. Maspoch, M. Porcel, Anal. Chim. Acta 431 (2001) 115–123.
[16] C. Hervás, A. Algar, M. Silva, J. Chem. Inf. Comput. Sci. 40 (2000) 724–731.
[17] D. Perez-Bendito, M. Silva, Trends Anal. Chem. 15 (1996) 232–240.
[18] Y. Zhang, H. Li, A. Hou, J. Havel, Talanta 65 (2005) 118–128.
[19] C. Hervás, A.C. Martínez, M. Silva, J.M. Serrano, J. Chem. Inf. Model. 45 (2005) 894–903.
[20] C. Hervás, M. Silva, J.M. Serrano, E. Orejuela, J. Chem. Inf. Comput. Sci. 44 (2004) 1576–1584.
[21] V. Dohnal, F. Zhang, H. Li, J. Havel, Electrophoresis 24 (2003) 2462–2468.
[22] T. Galeano-Diaz, A. Guiberteau, J.M. Ortiz, M.D. Lopez, F. Salinas, Chromatographia 53 (2001) 40–46.
[23] C. Hervás, R. Toledo, M. Silva, J. Chem. Inf. Comput. Sci. 41 (2001) 1083–1092.
[24] K.N. Gurney, Neural Netw. 5 (1992) 289–303.
[25] D.J. Janson, J.F. Frenzel, IEEE Expert 8 (1993) 26–33.
[26] A. Ismail, A.P. International Joint Conference on Neural Networks 2000, Proceedings of the Conference, Como, Italy, 2000, pp. 132–137.
[27] M. Schmitt, Neural Comput. 14 (2001) 241–301.
[28] P. Moscato, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimisation, McGraw-Hill, London, 1999, p. 219.
[29] A.G. Steenbeek, E. Marchiori, Proceedings of the IEEE International Conference on Evolutionary Computation ICEC'98, IEEE Press, New York, 1998, pp. 90–95.
[30] D. Whitley, S. Gordon, K. Mathias, in: Y. Davidor, H.P. Schwefel, R. Manner (Eds.), Parallel Problem Solving from Nature — PPSN III, Springer–Verlag, Berlin, 1994, pp. 6–15.
[31] C.R. Houck, J.A. Joines, M.G. Kay, Evol. Comput. 5 (1997) 31–60.
[32] J.A. Joines, M.G. Kay, in: R. Sarker, M. Mahamurdian, X. Yao (Eds.), Evolutionary Optimisation, Kluwer Academic Publishers, Boston, 2002.
[33] G. Vivo-Truyols, J.R. Torres-Lapasio, M.C. García-Alvarez-Coque, Chemom. Intell. Lab. Syst. 59 (2001) 89–106.
[34] G. Vivo-Truyols, J.R. Torres-Lapasio, A. Garrido-Frenich, M.C. García-Alvarez-Coque, Chemom. Intell. Lab. Syst. 59 (2001) 107–120.
[35] R. Jiménez-Prieto, M. Silva, D. Pérez-Bendito, Analyst 123 (1998) 1R–8R.
[36] R. Toledo, M. Silva, O. Khavrus, P.E. Strizhak, Analyst 125 (2000) 2118–2124.
[37] A.C. Martínez-Estudillo, C. Hervás-Martínez, F.J. Martínez-Estudillo, N. García-Pedrajas, IEEE T. Syst. Man Cy. Part B 36 (2006) 534–545.
[38] H. Bersini, B. Renders, First IEEE conference on evolutionary computation 1994, Proceedings of the Conference, Orlando, 1994, pp. 312–317.
[39] C.R. Houck, J.A. Joines, M.G. Kay, Comput. Oper. Res. 23 (1996) 587–596.
[40] P. Moscato, C. Cotta, in: F. Glower, G. Kochenberger (Eds.), Handbook of Metaheuristics, Kluwer Academic Publishers, Boston, 1999, pp. 1–56.
[41] P. Merz, B. Freisleben, Proceedings of the 1999 International Congress on Evolutionary Computation, 1999, pp. 2063–2070, Piscataway, NJ.
[42] P. Merz, B. Freisleben, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimisation, McGraw-Hill, London, 1999, pp. 245–260.

[43] P. Merz, B. Freisleben, Evol. Comput. 8 (2000) 61–91.

[44] P.J. Angeline, G.M. Saunders, J.B. Pollack, IEEE Trans. Neural Netw. 5 (1994) 54–65.

[45] X. Yao, Y. Liu, Y., IEEE Trans. Neural Netw. 8 (1997) 694–713.

[46] N. García-Pedrajas, C. Hervás-Martínez, J. Muñoz-Pérez, Neural Netw. 15 (2002) 1255–1274.

[47] S. Kirkpatrick, M.P. Vecchi, Science 220 (1983) 671–680.

[48] R.H.J.M Otten, L.P.P.P. van Ginneken, The Annealing Algorithm, Kluwer Academic Press, Boston, 1989.

[49] SPSS, Advanced Models. Copyright 12.0 SPSS Inc., Chicago, 2003.

[50] U.S. Environmental Protection Agency, Guidance on Choosing a Sampling Design for Environmental Data Collection, EPA, Washington, 2002.