J Math Chem (2015) 53:1080–1104
DOI 10.1007/s10910-015-0472-z

ORIGINAL PAPER

A robust hybrid algorithm (neural networks-AGDC) applied to non-isothermal kinetics of consecutive chemical reactions

Sonsoles Encinar · J. L. González-Hernández · M. Mar Canedo · Diana Juanes

Received: 26 October 2014 / Accepted: 5 January 2015 / Published online: 14 January 2015

Abstract This paper is concerned with the application of a Hybrid Algorithm (HA) to the determination of the Thermodynamic Activation Parameters (ATP) of a kinetic system of first order consecutive reactions. The 8 ATP's parameters involved in the Arrhenius and Eyring equations have been directly determined from the non-isothermal kinetic data without prior knowledge of the rate constants. AH is constituted by a combination of two algorithms based on different mathematical principles which are sequentially applied. In a first step, a "*soft modeling*" method of Artificial Neural Networks (ANN) is applied and the obtained values of ATP's parameters are used as *initial estimates* of a new optimization algorithm (AGDC) applied in a second stage to improve the values of the final parameters. The great success of HA is the efficient resolution of the ambiguity of the results obtained by ANN. In addition, comparing with the classic algorithms, which present the known weak points, HA offers important advantages: (a) the lack of necessity to know a priori the *initial estimates* since they are calculated from ANN application, (b) the low probability of being trapped at local minima, saddle points, etc. by means of the exhaustive control and suitable correction of the movement vector during the optimization process, and (c) the simultaneous determination of a higher number of parameters endowed with very different orders of magnitude.

Keywords Hybrid algorithm · Neural networks · AGDC · Non-isothermal kinetics · Thermodynamic activation parameters

S. Encinar · J. L. González-Hernández (✉) · M. M. Canedo · D. Juanes
Department of Physical Chemistry, Faculty of Chemistry, University of Salamanca, 37008 Salamanca, Spain
e-mail: jlgh93@usal.es

1 Introduction

The classic method to obtain the values of the Activation Thermodynamic Parameters (ATP) of a chemical reaction consists in the previous determination of the pairs of values of the rate constant k(T) from the isothermal kinetic experiences at different temperatures. Posteriorly, all pairs are fitted to straight lines by using Linear Regression, according the linearized Arrhenius and Eyring equations [1]. From each individual k(T) of each single reaction, we determine the values of the pre-exponential Factor (A) and the Activation Energy (E) from Arrhenius equation and the Activation Enthalpy ($\Delta H^{\neq}$) and the Activation Entropy ($\Delta S^{\neq}$) from the Eyring's equation, obtained from the Transition-State Theory (TST).

As for kinetic systems formed by "*r*" reactions with $k_r(T)$ rate constants, the procedure consists in the application of hard modeling methods in order to set up the system of ordinary differential equations (ODE). After solving them, the explicit mathematical expressions of the concentrations of all species involved in the kinetic system can be determined on the basis of $k_r(T)$ and time [2]. By using experimental data from a kinetic isothermal set carried out to different temperatures, the values of $k_r(T)$ can be calculated fitting the data to these explicit functions. Finally, the ATP's values are determined from the pairs of values k_r/T. This method leads to significant drawbacks: (a) a calculation of a high number of parameters is required, as it happens in our case of the consecutive reactions, where 8 ATP parameters have to be determined, (b) frequently, the ODE system doesn't provide exact mathematical solutions, so the $k_r(T)$ cannot be directly obtained and, therefore, ATP's values also remain unknown. Approaches (i.e. *Steady State* method) can be used to determine the values of the rate constants but the final ATP's values are not accurate at all, (c) a big amount of experimental work at the laboratory is needed to determine a reduced collective of value pairs $k_r(T)/T$, (d) it is also very common the necessity of knowing the values of other non kinetic magnitudes (i.e. equilibrium constants, etc.), which must be previously obtained by performing an additional task.

As a result of the foregoing, it is more convenient to design a new method to find directly the ATP's values without a previous determination of the rate constants. This new one requires to acquire a large amount of experimental information, as the data from the non-isothermal kinetics, which provide a more numerous set of data and final results more accurate. Therefore, in order to obtain the ATP' values, it is necessary to design a new algorithm of computational treatment, as the one proposed on the present work.

The computational determination of the parameters involved in mathematical functions is usually carried out by means of mathematical optimization methods applying numeric second order gradient algorithms [3]. These non-linear iterative fitting techniques are particularly used in Chemistry as well as other fields of Science. However, one of the most drawbacks of these optimization methods is their high sensitivity to the initial estimates of the supplied parameters. Only if these values are very close to the global minimum can a fast and reliable convergence of the iterative process be expected, guaranteeing success of the parameter optimization. If the initial estimates are far from the global minimum the process may become divergent or reach a singular point ("local minimum", "saddle point", etc.), leading the optimization process to fail.

This often happens in the treatment of kinetic models when the number of parameters to be optimized is moderately high. Additionally there is a series of problems involving identifiability and distinguishability [4,5], which lead to different types of ambiguity in the solutions to the stiff systems of ordinary differential equations (ODE). In light of this, it would seem appropriate to design and apply a new method that, initially, would provide an approach to the global optimum and then use such results as a starting point to apply a robust gradient method that will guarantee the success of the mathematical optimization of parameters.

In the present work we designed and successfully applied to the kinetic system of consecutive reactions, a new Hybrid Algorithm (HA) able to determine the ATP's values directly from non-isothermal kinetic data without the need of determining the rate constants in a previous step.

A HA is the one in which 2 or more different algorithms are combined in order to solve the same mathematical problem. Depending on data, the HA can perform by means of choosing the algorithm that fits better the data or switching between them during the whole application process. This second way corresponds to the HA proposed in this work. The final objective is to reach the combination of the most appropriate characteristics of each one, so the general overall HA can perform better than individual components. The HA refers not only to a combination of multiple algorithms to solve a different problem. In fact, many algorithms can be considered themselves as combinations of simple pieces. HA is a combination of algorithms that solve the same problem but differ in their characteristics, particularly in efficiency.

The HA proposed in this work comprises two algorithm based on different mathematical principles in which sequential application is carried out. A "soft modeling" method is applied by using Artificial Neural Networks (ANN) [6], which presents the valuable advantage of making the use of initial estimates of the parameters unnecessary. The final objective is to determine the parameter values (outputs) in the neighborhood of the optimum global. These values have been used as the initial estimates values of a robust and efficient numeric second order gradient algorithm (AGDC) [7], able to reach the desired global minimum to guarantee the success of the final optimization of the parameters. The application of HA to the kinetic system of consecutive reactions provides several important advantages since the direct optimization of those parameters became possible, with no need of calculating the rate constants $k_r(T)$ in previous steps. It must be considered that 8 ATP parameters are involved in the studied kinetic system of consecutive reactions, representing a very elevated number of parameters to be determined. The treatment requires to acquire the kinetic data from non-isothermal kinetic experiments, imposing a controlled variation of temperature along the reaction kinetic. The number of the set of kinetic data computed in the classic isothermal procedure is very small since it is limited by experimental conditions, when dealing with a low number of isothermal experiments (normally, 10–12), each of them performed at a different temperature. In the case of non-isothermal experiments, a single replicate kinetic experiment is enough since it allowed the computation of a huge set of kinetic data which, bearing in mind the laboratory time and the amount of reagents saved, is a great and important advantage.

The literature contains several recent references of the AGDC mathematical optimization algorithm that is a symbolic second-order gradient method that performs a rigorous analysis and control of the movement vector and of each of its terms. The numeric version of this algorithm was designed and successfully implemented at our laboratory within the field of Chemistry in the treatment of many systems with kinetic, analytical and thermodynamic purposes. Thus computational applications have developed for the quantitative analysis of static (SMM) and dynamic (DMM) multicomponent mixtures [7,8] or for the determination of kinetic constants in different reaction mechanisms after the application of the numerical computational program KINMODEL(AGDC) [9]. In this type of joint optimization of several parameters the problem of ambiguity in the solution is analyzed in reaction mechanisms comprising first-order reactions since it is common to find several (two or more) groups of parameters that fit the experimental kinetic data. The AGDC algorithm has also been used for the determination of the initial concentrations of the species involved in the reaction mechanism, individually or jointly, with the kinetic constants of the elementary reactions comprising the mechanism [10]. In this paper there is analyzed the influence of a series of factors affecting the optimization process and in addition there is studied the possibility of the existence of ambiguity in the solutions, since parameters of different natures and orders of magnitude are determined. Recently a novel *"hard-modeling"* computational approach, based in the algorithm AGDC(MW) [11], is proposed for the simultaneous optimization of the rate constants and molar absorption coefficients. The approach is the result of combining 3 different strategies based in 3 versions of AGDC and an evident improvement in the values of the optimized parameters has been obtained in comparison with the application of a single strategy. Several authors have applied Hybrid Algorithms (HA) in the field of Chemistry in the literature that essentially comprises Genetic Algorithms (GA). Their design and application are currently able to solve problems pending solution and reveal the high degree of reliability and precision in the results obtained. Of interest is one review [12] in which the authors consider a broad range of applications of different types of GA in Chemometrics: Maeder et al. [13] determine the rate and equilibrium constants of reaction mechanisms by application of a HA based on a GA. Hervás et al. [14] use GA and pruning computational neural networks for selecting the number of inputs required to correct temperature variations in kinetic-based determinations. Artificial Neural Networks offer a versatile "*soft modeling*" method that can be applied in diverse fields with acceptable results [15]. The method is applied for quantitative purposes, among others, in the so-called *Principal Component Analysis* (PCA) method, in which there are no explicit functions of multivariate correlation or, if there are, they are extraordinarily complex. The application of ANN (alone or combined with a mathematical optimization algorithm) in chemical kinetics when a single chemical reaction is considered, [16–18] can provide acceptable results. However, it is necessary to perform an exhaustive process of *training* of the neural network in order to obtain its optimal architectures, meaning that the method is time-consuming and tedious and that it cannot be used individually in all cases. Nevertheless, it is an ideal method for carrying out the approximation of the ATP's values to those of the global minimum, without previously knowing anything about the magnitude and sign of the parameters.

2 Theoretical aspects

2.1 Kinetic and thermodynamic aspects in non-isothermal kinetics

Let us consider in general a chemical system formed by n_r chemical elementary reactions where n_s chemical species can be involved. According to IUPAC's norms [19], the r-th chemical reaction can be expressed for the generic equation

$$0 = \sum_{j=1}^{n_s} \nu_{j,r} B_j \tag{1}$$

where, B_j represents the chemical species involved in the system of reactions; $r = (1, \ldots, n_r)$, the number of chemical reactions; $j = (1, \ldots, n_s)$, the number of chemical species; $\nu_{j,r}$, the stoichiometric coefficient of the species B_j in the r-th reaction; $\nu_{j,r} < 0$ when B_j plays only the role of reactant in the r-th reaction and $\nu_{j,r} > 0$ when B_j plays only the role of product in the r-th.

When the reaction is an elementary or concerted one, the absolute values of the kinetic order ($z_{l,r}$) and stoichiometric coefficient of B_j coincide, that is $|\nu_{l,r}| = |z_{l,r}|$. The rate differential equation of the chemical species B_j in the r-th is given by

$$d\left[B_j\right]/dt = k_r \nu_{j,r} \prod_{l=1}^{n_s} [B_l]^{|z_{l,r}|} \tag{2}$$

where B_l are the species playing only the role of reactants in the r-th reaction ($\nu_{l,r} < 0$) and k_r is the kinetic rate constant of the r-th reaction. Each chemical species can take part in several reactions and the rate differential equations will be the sum extended over those reactions where the reactant B_l appears, obtaining a system of ordinary differential equations (ODE) according to the generic equation,

$$\mathrm{d}\left[\mathrm{B_j}\right]/\mathrm{dt} = \sum_{\mathrm{r}=1}^{\mathrm{n_r}} \mathrm{k_r} \nu_{j,r} \prod_{\mathrm{l}=1}^{\mathrm{n_s}} [\mathrm{B_l}]^{|z_{\mathrm{l,r}}|} \tag{3}$$

The general solution of the system of rate ODE give the explicit function of the concentrations of the all species with time ($[B_j]_{t_i}$). If the experimental data are expressed in absorbance, we have to consider the Lambert–Beer–Bouguer law:

$$A^{\lambda}_{j,t_i} = \varepsilon^{\lambda}_j \cdot [B_j]_{t_i} \tag{4}$$

where A^{λ}_{j,t_i} is the absorbance of the species B_j at the, time t_i and path length 1 cm and ε^{λ}_j is the molar absorption coefficient of Bj at the wavelength λ. The absorbance of the mixture (A^{λ}_{T,t_i}) measured at wavelength λ, time t_i and temperature T and, can be expressed as:

$$A^{\lambda}_{T,t_i} = \sum_{j=1}^{n_s} A^{\lambda}_{j,t_i} = \sum_{j=1}^{n_s} \varepsilon^{\lambda}_j \cdot [B_j]_{t_i} \tag{5}$$

The chemical system formed by 2 first order consecutive reactions has been studied in the present work and can be represented according to Eq. (1), as

$$\nu_{1,1} B_1 \overset{k_{12}}{\rightarrow} \nu_{2,1} B_2$$
$$\nu_{2,2} B_2 \overset{k_{23}}{\rightarrow} \nu_{2,3} B_3$$

Considering $\nu_{1,1} = \nu_{2,2} = -1$ and $\nu_{2,1} = \nu_{2,3} = 1$, we have the first order consecutive reactions system studied in this work.

$$B_1 \overset{k_{12}}{\rightarrow} B_2 \overset{k_{23}}{\rightarrow} B_3$$

According this, the system of ordinary differential equations (ODE) can be expressed using matrix notation as

$$d/dt \begin{vmatrix} [B_1] \\ [B_2] \\ [B_3] \end{vmatrix} = \begin{vmatrix} -k_{12}(T) & 0 & 0 \\ k_{12}(T) & -k_{23}(T) & 0 \\ 0 & k_{23}(T) & 0 \end{vmatrix} \begin{vmatrix} [B_1] \\ [B_2] \\ [B_3] \end{vmatrix} \tag{6}$$

2.1.1 Solution of the first differential equation of the ODE system

The first rate differential equation of the ODE system (6) is

$$d[B_1]/dt = -k_{12}\,(T)\,[B_1] \tag{7}$$

that can be individually solved taken in account that only depends of $[B_1]$. It is convenient to consider here the general kinetic treatment of a chemical reaction in non isothermal conditions.

$$\nu_1\, B_1 \rightarrow Products \tag{8}$$

The rate differential equation to this reaction, expressed in function of the *extent of reaction* variable in units of molar concentration (ξ_1') when the reactant B_1 obey to a first order kinetic, can be written as:

$$d\xi_1'/dt = k_{12}\,(T)\,([B_1]_0 - |\nu_1|\,\xi_1') \tag{9}$$

where $[B_1]_0$ is the initial concentration of B_1 species and $k_{12}\,(T)$ is the rate constant, a function of T in non-isothermal conditions. Separating variables and after integration of the first member of the equation, we have,

$$-\,(1/|\nu_1|)\,ln\left\{\left([B_1]_0 - |\nu_1|\,\xi_1'\right)/[B_1]_0\right)\right\} = \int_0^t k_{12}\,(T)\,dt \tag{10}$$

The Activation Energy, (E_{12}) and the pre-exponential Factor (A_{12}) can be determined substituting $k_{12}(T)$ in Eq. (10) according the Arrhenius equation, where α_1 is the remaining molar fraction of the reactant B_1.

$$-(1/|\nu_1|)\ln\alpha_1 = \int_0^t A_{12}e^{-E_{12}/RT}dt \tag{11}$$

In non-isothermal conditions, the second member of Eq. (11) cannot be integrated since there are two dependent variables [T = f(t)]. It will be crucial to establish the identity of this function because the mathematical method of resolution of the Eq. (11) will be different. However 2 requirements must be met: the function must be monotonically increasing, since is this way it will be possible to minimize the great differences in the reaction rate existing between the beginning and the end of the reaction; the rate of heating must be suitable for the interval of time studied and its profile must be reproducible in the laboratory. In order to choose a variation type T/t we consider two possibilities for the function:

1. hyperbolic branch

$$1/T = (1/T_0) - mt \tag{12}$$

which leads to the following equation:

$$-(1/|\nu_1|)\ln\alpha_1 = \int_0^t A_{12}e^{-E_{12}/R[(1/T_0)-mt]}dt \tag{13}$$

This hyperbolic function has *a priori* a great advantage since the primitive function of the integral of the Eq. (13) exists being the mathematical exact solution the following expression:

$$\alpha_1 = e^{-[(|\nu_1|A_{12}R/mE_{12})\left(e^{-E_{12}/RT_0}\right)\left(e^{(mE_{12}/R)t}-1\right)]} \tag{14}$$

However 2 considerations must be done: (i) the explicit function of $\boldsymbol{\alpha}_1$ depending on the time (14) is not simple and there is necessary the application of a method of sufficiently robust treatment for the determination of the Thermodynamic parameters E_{12} and A_{12}. In addition it is necessary to consider the great difference in the order of magnitude of both parameters that complicates extraordinarily the success in the application of the method of treatment. (ii) the experimental points (T/t) must satisfy to this hyperbolic function and therefore, it is necessary to reproduce the profile of the curve of the function (T/t) (12) with the points obtained in the laboratory.

2. function of polynomial type of n-th degree

$$T = \sum_{j=0}^{j=n} a_j t^j \tag{15}$$

or even, we can propose a first order polynomial, that is, a lineal function ($T = T_0 + a_0\, t$)

$$-(1/|\nu_1|)\ln\alpha_1 = \int_0^t A_{12} e^{-E_{12}/R[(1/\sum_{j=0}^{j=n} a_j t^j)]} dt \tag{16}$$

This equation does not have mathematical exact solution since the integration cannot be performed. We have 2 options of treatment: (a) performing a numerical integration of the Eq. (13) using appropriate *quadrature* formulas applying a suitable numerical algorithm for the resolution of integrals (Simpson, Lobatto, Gauss-Kronrod, Vectorized, etc.), and (b) numerical resolution directly from the beginning of the ordinary differential equation (7) expressed in terms of Arrhenius's equation, that is:

$$d[B_1] = -[B_1]\, A_{12} e^{-E_{12}/RT} dt \tag{17}$$

The solution of this differential equation can be performed by means of the application of numerical methods that they must be suitable for the treatment of *stiff* systems (Runge–Kutta methods, Gear's method (BDFs), Rosenbrock formula, trapezoidal rule with "free" interpolant…etc).

In the case of the determination of the ATP's parameters involved in the Eyring's equation ($\Delta H_{12}^{\neq}$ and $\Delta S_{12}^{\neq}$) the mathematical treatment of the differential rate Eq. (10) is different with respect to the case of Arrhenius. It can be expressed as:

$$-(1/|\nu_1|)\ln\alpha_1 = \int_0^t (k_B/h) T e^{-\Delta H_{12}^{\neq}/RT} e^{\Delta S_{12}^{\neq}/R} dt \tag{18}$$

The solution of the integral in the second member of (18) depends of the type of the explicit function $T = f(t)$ used. We discard the inverse function of the temperature corresponding to a hyperbolic branch (12) because in this case is not possible a mathematical exact solution of the integral. Therefore, we consider exclusively the polynomial function of n-th degree (15) and the treatment to be performed is identical than the case of Arrhenius's equation. We consider 2 options: numerical integration of the equation using appropriate *quadrature formulas* and the numerical resolution directly from the beginning of the (ODE) (18), by means of the application of methods suitable for the treatment of stiff systems.

$$d[B_1] = -[B_1](k_B/h) T e^{-\Delta H_{12}^{\neq}/RT} e^{\Delta S_{12}^{\neq}/R} dt \tag{19}$$

2.1.2 Solution of the other ODE system differential equations

The remaining differential equations that constitute the system (6) for the substances B_2 and B_3, are:

$$d[B_2]/dt = k_{12}(T)[B_1] - k_{23}(T)[B_2] \tag{20}$$

$$d[B_3]/dt = k_{23}(T)[B_2] \tag{21}$$

which resolution in isothermal conditions lead to the solutions:

$$[B_2] = \frac{[B_1]_0\, k_{12}}{k_{23} - k_{12}} \left(e^{-k_{12}t} - e^{-k_{23}t} \right) \tag{22}$$

$$[B_3] = [B_1]_0 \left(1 - \frac{k_{23}}{k_{23} - k_{12}} e^{-k_{12}t} + \frac{k_{12}}{k_{23} - k_{12}} e^{-k_{23}t} \right) \tag{23}$$

However, if non-isothermal conditions are considered and after substituting $k_{12}\,(T)$ and $k_{23}\,(T)$ on Eqs. (20) and (21) based on the Arrhenius and Eyring's functions, these differential equations do not provide us an exact mathematic solution. In conclusion, there is non-explicit functions of $\alpha_2(t_i)$ and $\alpha_3(t_i)$ depending of ATP parameters and time. The solutions are exclusively numeric ones, what it means only discrete values of $\alpha_2(t_i)$ and $\alpha_3(t_i)$ can be determined for each time value. For that reason, it is necessary to determine $k_{12}(T_i)$ y $k_{23}(T_i)$ for each value of T_i, by substituting the t_i values on the following equations:

$$\begin{aligned} k_{12}\,(T_i) &= A_{12} e^{-E_{12}/R \sum_{j=0}^{j=n} a_j t_i^j} \\ &= (k_B/h) \sum_{j=0}^{j=n} a_j t_i^j e^{-\Delta H_{12}^{\neq}/R \sum_{j=0}^{j=n} a_j t_i^j} e^{\Delta S_{12}^{\neq}/R} \end{aligned} \tag{24}$$

$$\begin{aligned} k_{23}\,(T_i) &= A_{23} e^{-E_{23}/R \sum_{j=0}^{j=n} a_j t_i^j} \\ &= (k_B/h) \sum_{j=0}^{j=n} a_j t_i^j e^{-\Delta H_{23}^{\neq}/R \sum_{j=0}^{j=n} a_j t_i^j} e^{\Delta S_{23}^{\neq}/R} \end{aligned} \tag{25}$$

It is important to consider that the values of A_{12}, E_{12}, $\Delta H_{12}^{\neq}$ and $\Delta S_{12}^{\neq}$ are previously known from the application of ANN method. These values are involved in Eq. (24) and they must be constant in the later process of determination and optimization of A_{23}, E_{23}, $\Delta H_{23}^{\neq}$ and $\Delta S_{23}^{\neq}$ by means of complete application of HA to Eq. (25). Finally the numeric values of $\alpha_2(t_i)$ y $\alpha_3\,(t_i)$ are determined by the Eqs. (22) and (23).

$$\alpha_2(t_i) = \frac{k_{12}(T_i)}{k_{23}(T_i) - k_{12}(T_i)} \left(e^{-k_{12}(T_i)t_i} - e^{-k_{23}(T_i)t_i} \right) \tag{26}$$

$$\alpha_3(t_i) = \left(1 - \frac{k_{23}(T_i)}{k_{23}(T_i) - k_{12}(T_i)} e^{-k_{12}(T_i)t} + \frac{k_{12}(T_{t_i})}{k_{23}(T_{t_i}) - k_{12}(T_{t_i})} e^{-k_{23}(T_i)t_i} \right) \tag{27}$$

These 2 sets of kinetic value pairs $[\alpha_2(t_i)/t_i]$ and $[\alpha_3(t_i)/t_i]$ are going to constitute the data of the *input curves* for their treatment with the HA, which it will provide us all the ATP's values A_{12}, A_{23}, E_{12}, E_{23}, $\Delta H_{12}^{\neq}$, $\Delta H_{23}^{\neq}$, $\Delta S_{12}^{\neq}$ and $\Delta S_{23}^{\neq}$.

2.2 Hybrid Algorithm: HA (ANN-AGDC)

The application of *Hybrid Algorithm (HA)* to kinetic system of consecutive reactions has been carried out by 2 different procedures, depending if it is implemented into the

resolution of the 1st differential equation (7) from the 3 equations of the ODE system (6), or into the resolution of the other 2 differential equations of the system:

I. Resolution of the 1st differential equation of the ODE system (6).

The treatment of α_1 /t data pairs is accomplished in order to determine the ATP parameters's A_{12}, E_{12}, $\Delta H_{12}^{\neq}$ and $\Delta S_{12}^{\neq}$, by following 2 different steps:

(1) Method ANN. The input data are constituted by a matrix of curves (α_1 /t) of synthetic kinetic data endowed with *"noise"* and 2 *targets* matrices, which contain the ATP's values for each curve consistently distributed on the basis of the appropriate *Experimental Design*. The results are 2 *outputs* matrices composed by the ANN obtained values of the former 4 ATP parameters, which will be the *initial estimates* for the following step.

(2) AGDC algorithm. The values of the 4 ATP parameters are optimized by implementing the numeric algorithm AGDC adapted to the considered cases and depending on the type of T/t function chosen. The matrix of curves (α_1 /t) is used as *input data* and the 4 ATP parameters, determined by the previous ANN method application, as *initial estimates*. The final solutions obtained by the HA application are the optimized values of the 4 parameters A_{12}, E_{12}, $\Delta H_{12}^{\neq}$ and $\Delta S_{12}^{\neq}$.

II. Resolution of the other 2 differential equations of the ODE system

The treatment of α_2 /t and α_3 /t data pairs is performed in order to determine all the ATP parameters. The whole procedure comprises 2 steps:

(1) Method ANN. The *input data* is constituted by the curves' matrix of the kinetic pairs α_2 /t and α_3 /t of synthetic data with *noise* and the 2 *targets* matrices constituted by the values of ATP's A_{12}, E_{12}, A_{23}, E_{23} (Arrhenius) *and* $\Delta H_{12}^{\neq}$ $\Delta S_{12}^{\neq}$, $\Delta H_{23}^{\neq}$, $\Delta S_{23}^{\neq}$ (Eyring), respectively. These 2 matrices contain 2 types of ATP parameters: (a) the ones with subscripts $_{(12)}$, which values were optimized by the application results from procedure I and will remain constant and invariable (24) along the steps of the procedure II. (b) the ones with subscripts $_{(23)}$, which values will be conveniently modified—Eq. (25) for the determination from ANN and the optimization by AGDC. The solutions are 2 *outputs* matrices that contain the values of ANN determined from the 2 sets of 4 ATP Arrhenius and Eyring parameters (8 in total) and will be the *initial estimates* for the following step of the process.

(2) AGDC algorithm. The numeric algorithm AGDC, is applied depending on the kind of T/t function in order to optimize the 4 ATP parameters with subscripts $_{(23)}$,that is, A_{23}, E_{23} $\Delta H_{23}^{\neq}$, $\Delta S_{23}^{\neq}$ By way of explanation, the AGDC iterative process starts up with the *initial estimates* values of the parameters. Although the 4 ATP parameters with subscripts $_{(12)}$ are not modified, the other 4 ATP ones with subscripts $_{(23)}$ are determined and optimized by the ANN and AGCD sequential processes. The final solutions after application of the HA are the values of 8 ATP parameters (A_{12}, A_{23}, E_{12}, E_{23}, $\Delta H_{12}^{\neq}$, $\Delta H_{23}^{\neq}$ $\Delta S_{12}^{\neq}$ and $\Delta S_{23}^{\neq}$).

2.2.1 Artificial neural networks

Artificial Neural Networks are parallel interconnected networks of simple computational elements called neurons and are structured in layers that are intended to interact with the objects of the real world in a similar way to the biological nervous systems

[20]. Parallel processing is the ability of the brain to simultaneously process incoming stimuli of differing quality. The multilayer neural network uses sets of input data and parameters (called *targets*), distributed in 2 input matrices when Matlab [21] is applied. The elements of the input matrix are the calculated synthetic values, where one row contains a single curve of the data and all the curves thus obtained (n_c) are grouped in an *input data* matrix. The *target matrix* is formed by the sets of parameters (n_p). In our case, the input data matrix contained the kinetic data of all curves (α_j, A_T, A_j, $[B_j]$) and the *target matrix* ($n_c \times n_p$) contained the set of kinetic rate constants (k_{mn}). Formally, a multilayer neural network is an oriented graph in which the nodes represent a set of processing units, called neurons, and the connections represent the information flow channels. Each connection between two neurons has an associated value called *weight* which specifies the strength of the connection between neurons. Positive and negative values determine excitatory and inhibitory connections, respectively. The choice of a specific class of networks for the approximation of a nonlinear map depends on a variety of factors dictated by the context and is related to the desired accuracy and the prior information available concerning the input–output pairs.

The first layer of a multilayer neural network contains neurons that receive the input data values from the elements of the input data matrices. This information is transmitted from the i-th neuron of a layer to the j-th neuron of the subsequent one, with a weight w_{ji}. A neuron parameter (*bias*) is summed with the weighted inputs of the neurons and passed through the transfer function to generate the output of the neurons.

The layer following the input one is called *hidden* In each neuron of a *hidden* layer the weighed inputs coming from the previous one are summed with each other and added to a *bias*. The result is then transformed by means of a suitable mathematical function to obtain an output called *activation of the neuron*, which is transferred to the neurons in the next layer after another weighing step. The output parameters values are calculated in the last layer (*output* layer) by means of a suitable transformation function.

The process described is called to as the *training* or*learning* of the multilayer neural network and constitutes an iterative method where the iterations are called *epochs*. After each *epoch,* the calculated values of the parameters are grouped in the *output matrix* (b_{ij}^{output}) and they are compared with those of the corresponding curve in the *target matrix* (b_{ij}^{target}) and the optimum value of the Mean Squared Error (MSE), expressed in absolute value, is calculated according Eq. (28)

$$MSE = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} (b_{ij}^{output} - b_{ij}^{target})^2}{n_p \cdot n_c} \right)^{1/2} \tag{28}$$

where n_c is the input number of curves and n_p is the number of parameters, $n_c \times n_p$ being the dimensions of both matrices (*output matrix* and *target matrix*).

During the process of *training, weights* and *bias* values are modified with suitable mathematical optimization algorithms in order to minimize the calculated values of MSE in each *epoch*. In the present work, the *back-propagation* algorithm was used.

The iterative process finishes when the minimum value of MSE is reached, after which the *training* process can be considered to be completed.

It is necessary to know the optimal architecture and topology of the multilayer neural network in order to obtain the best results when ANN is applied to the system under study. We have used a method of *trial and error* by minimizing the MSE values obtained for the different possible configurations of the same number of *hidden* layer/s. It must to determine the minimum value (optimum) of the MSE for all possible configurations (i.e. in the range between 1 and 12) for the *hidden* layer/s chosen. For each *hidden* layer, a graph of MSE values versus the number of neurons shows that initially, for the lower configurations, the value of the MSE decreases rapidly when the number of neurons increases, but after a constant value or a poor improvement is obtained. The optimum number of neurons (*configuration*) in that *hidden* layer is given by the point of intersection of the two branches of the graph. Sometimes, a small minimum appears near this intersection point. The architecture of the neural network can be written in abbreviated notation as (n_{inp}, n_{hid}, n_{out}), where n_{inp} is the number of neurons in the *input* layer, n_{hid} in the *hidden* layer and n_{out} in the *output* layer.

Neural network *training* is completed with the processes of *validation* and *testing* reach satisfactory results. These are 2 control and verification processes of the iterative minimization method between the elements of the *output* and *target* matrices. Among the different curves comprising the *input* matrix, random choice is made of a percentage of the total, established previously (5, 10 %. . .), which gives rise to a "*sub-matrix*" of input curves that are subjected to iterative optimization until a minimum MSE value is reached. It is thus possible to verify the validity of the *training* process by ensuring that it is convergent, that it has an appropriate termination, and that there not been any *overfitting*, since any possible *overtraining* has taken been into account. *Validation* process is completed when in a given number ($\geq$6) of consecutive *epochs* the MSE remains constant or shows a slight tendency to increase. The *testing* process is similar, except that the control for terminating the process is performed by controlling the computation time instead of the number of *epochs*. The process of *prediction* consists of the determination of the unknown parameters from a set of experimental data after application of the optimal and trained neural network. Obviously, the elements of the *target* matrix are unknown for this *prediction* process, and only the *input data* matrix is provided to the neural network. The elements of the *input data* matrix in the process of *prediction* will be experimental kinetic values (α_j, A_T, A_j, $[B_j]$, etc.) acquired from a kinetic system of reactions developed at the laboratory.

2.2.2 AGDC algorithm

The AGDC is an algorithm of mathematical optimization based on a second-order gradient method that minimizes, by means of an iterative process, the numerical function SQD:

$$\mathrm{SQD}\,(\mathbf{X}) = \sum_{i=1}^{N_d} \left(\left(\alpha_j\,(t_i)\right)_C - \left(\alpha_j\,(t_i)\right)_E \right)^2 \tag{29}$$

$$\alpha_j\,(t_i) = \left[B_j\right]_i / \left[B_1\right]_0 \tag{30}$$

where **X** is the vector that contains the parameters to be optimized. In this case the components of the **X** vector can be pairs of values of the Activation Thermodynamic parameters involved IN a first order consecutive reactions system (E_{12} and A_{12}, $\Delta S_{12}^{\neq}$ and $\Delta H_{12}^{\neq}$, E_{23} and A_{23}, $\Delta S_{23}^{\neq}$ and $\Delta H_{23}^{\neq}$). The AGDC algorithm first uses, as the movement vector, the one indicated by the Gauss–Newton method [3,22]:

$$\boldsymbol{p}^{(m)} = -\boldsymbol{g}^{(m)} \left[\boldsymbol{H}^{(m)}\right]^{-1} \tag{31}$$

where $\mathbf{p}^{(m)}$, $\mathbf{g}^{(m)}$ and $[\mathbf{H}^{(m)}]^{-1}$ are respectively the movement vector, the gradient vector and the inverse of the Hessian matrix of the iteration m, whose terms are derived from the function to be minimized (SQD) with respect to each of the parameters to be determined (**X**).

If the residuals are given by:

$$RES_i = \left(\alpha_j\right)_C - \left(\alpha_j\right)_E \tag{32}$$

then $\mathbf{g}^{(m)}$ and $\mathbf{H}^{(m)}$ are given by:

$$\mathbf{g} = 2 \begin{bmatrix} \sum_{i=1}^{N_d} RES_i \frac{\partial (\alpha_j)_C}{\partial X_1} \\ \sum_{i=1}^{N_d} RES_i \frac{\partial (\alpha_j)_C}{\partial X_2} \end{bmatrix} \tag{33}$$

$$\mathbf{H} = 2 \begin{bmatrix} \sum_{i=1}^{N_d} \left(\frac{\partial (\alpha_j)_C}{\partial X_1}\right)^2 & \sum_{i=1}^{N_d} \left(\frac{\partial (\alpha_j)_C}{\partial X_1}\right)\left(\frac{\partial (\alpha_j)_C}{\partial X_2}\right) \\ \sum_{i=1}^{N_d} \left(\frac{\partial (\alpha_j)_C}{\partial X_2}\right)\left(\frac{\partial (\alpha_j)_C}{\partial X_1}\right) & \sum_{i=1}^{N_d} \left(\frac{\partial (\alpha_j)_C}{\partial X_2}\right)^2 \end{bmatrix} \tag{34}$$

Calculation of the derivatives is performed numerically by means of the central differences method [23] and the inverse of the Hessian matrix is computed by means of the MatLab application [21] that computes the inverse of a square matrix using LU factorization.

The AGDC algorithm performs a rigorous analysis and control of the movement vector and of each of its terms, and suitable modifications can be made if any errors are detected, thereby ensuring successful optimization. Once the optimization process has been achieved, the program determines the errors of the optimized parameters [24] and performs an exhaustive analysis of the residuals thus allowing the goodness of fit to be checked [25].

Schematically the AGDC algorithm can be written as follows:

Iteration m → Input data: $\mathbf{X}^{(m)}$, $\mathrm{SQD}^{(m)}$.

1. Calculate the vector of movement.
   1.1. Compute partial numerical derivatives of $(\alpha_j)_C$ with respect to the parameters to be determined $\boldsymbol{X}_\mathbf{p}$, $\left(\partial \left(\alpha_j\right)_C / \partial X_p\right)^{(m)}$.
   1.2. Compute Gradient vector and Hessian Matrix ($\mathbf{g}^{(m)}$ and $\mathbf{H}^{(m)}$). (33), (34).

1.3. Compute $(\mathbf{H}^{(m)})^{-1}$.
1.4. Calculate the components of the vector of movement $\mathbf{p}^{(m)}$ (31).
2. Control and correction of the direction of the vector of movement $\mathbf{p}^{(m)}$.
2.1. If $\mathbf{H}^{(m)}$ is singular, $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$, go to 3.
2.2. If $\mathbf{p}^{(m)}\,\mathbf{g}^{(m)} < \varepsilon$ (ε = scalar close to zero), $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$ and go to 3.
2.3. If $\mathbf{p}^{(m)}\,\mathbf{g}^{(m)} > 0$, $\mathbf{p}^{(m)} = -\mathbf{p}^{(m)}$.
3. Control the length of the vector of movement $\mathbf{p}^{(m)}$.
3.1. Compute the scalar $(\lambda^{(m)})$ by the method of Hartley [3].
3.2. $\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} + \lambda^{(m)}\mathbf{p}^{(m)}$.
3.3. Determinate the $\mathrm{SQD}^{(m+1)}$ function (29).
3.4. If the Goldstein–Armijo *criterium* [22] is satisfied go to 4.
3.5. $\lambda^{(m)} = \lambda^{(m)}/2$ go to 3.2.
4. Calculate:

$$CON = \left| \frac{SQD^{(m+1)} - SQD^{(m)}}{SQD^{(m)}} \right| \tag{35}$$

5. If convergence is not attained (CON > CC), set m = m + 1 and go to 1.
6. $\mathbf{X}^{(m+1)}$ = Optimized Parameters.
7. END optimization.

The procedure followed to carry out the optimization of the activation parameters is schematically expressed as follows:
0. Input data: Experimental data of $(\alpha_j)_E/t_i$, $[B_j]_0$, Convergence Criteria(CC), Initial estimates of the unknown parameters $\mathbf{X}^{(0)}$ $\left(E_{12}^{(0)},\ A_{12}^{(0)},\ \Delta S_{12}^{\neq(0)},\ \Delta H_{12}^{\neq(0)},\ E_{23}^{(0)},\ A_{23}^{(0)},\ \Delta S_{23}^{\neq(0)},\ \Delta H_{23}^{\neq(0)}\right)$ (values of *outputs* from ANN application).

I. **Optimization of: $\mathbf{E_{12}}$, $\mathbf{A_{12}}$, $\mathbf{\Delta S_{12}^{\neq}}$ and $\mathbf{\Delta H_{12}^{\neq}}$.**
Ia. $\mathbf{X} = [E_{12},\ A_{12}] \cdot m = 0.\ \mathbf{X}^{(0)} = \left[E_{12}^{(0)},\ A_{12}^{(0)}\right]$.
Ia1. Calculate $(\alpha_1^{(0)})_C$ (30).
(i) T = f(t) inverse hyperbolic (12)→ Mathematical exact solution (14).
(ii) T = f(t) polynomial of n-th degree (15) → Numerical integration of (16) or Numerical solution of the rate differential equation (17).
Ia2. Calculate $\mathrm{SQD}^{(0)}$ (29).
Ia3. **AGDC Algorithm.**
Ia4. Optimized parameters: $\mathbf{X}^* = [E_{12},\ A_{12}]$ → Calculate the errors of E_{12}, A_{12}.
Ib. $\mathbf{X} = \left[\Delta S_{12}^{\neq},\ \Delta H_{12}^{\neq}\right].m = 0.\ \mathbf{X}^{(0)} = \left[\Delta S_{12}^{\neq(0)},\ \Delta H_{12}^{\neq(0)}\right]$.
Ib1. Calculate $(\alpha_1^{(0)})_C$ (30): T =f (t) Polynomial of n-th degree (15)→ Numerical integration (18) or Numerical solution of the rate differential equation.
Ib2. Calculate $\mathrm{SQD}^{(0)}$ (29).
Ib3. **AGDC Algorithm.**
Ib4. Optimized parameters: $\mathbf{X}^* = \left[\Delta S_{12}^{\neq},\ \Delta H_{12}^{\neq}\right]$ → Calculate the errors of $\Delta S_{12}^{\neq}\Delta H_{12}^{\neq}$.

II. **Optimization of: $\mathbf{E_{23}}$, $\mathbf{A_{23}}$, $\mathbf{\Delta S_{23}^{\neq}}$ and $\mathbf{\Delta H_{23}^{\neq}}$.**

IIa. $\mathbf{X} = [\mathrm{E_{23}},\ \mathrm{A_{23}}] \cdot m = 0.\mathbf{X}^{(0)} = \left[\mathrm{E}_{23}^{(0)},\ \mathrm{A}_{23}^{(0)}\right]$.

IIa1. Calculate $\left(\alpha_2(\mathrm{t_i})^{(0)}\right)_{\mathrm{C}}$, $\left(\alpha_3(\mathrm{t_i})^{(0)}\right)_{\mathrm{C}}$(30).

(i) Calculate $T_i = \sum_{j=0}^{j=n} a_{j,i} t_i^j$

(ii) Calculate $k_{12}\,(T_i) = A_{12} e^{-E_{12}/R \sum_{j=0}^{j=n} a_j t_i^j}$

(iii) Calculate $k_{23}\,(T_i) = A_{23}^{(0)} e^{-E_{23}^{(0)}/R \sum_{j=0}^{j=n} a_j t_i^j}$

(iv) Numerical solution of the rate differential equations (20), (21).

IIa2. Calculate $\mathrm{SQD}^{(0)}$ (29).

IIa3. **AGDC Algorithm.**

IIa4. Optimized parameters: $\mathbf{X}^* = [\mathrm{E_{23}},\ \mathrm{A_{23}}]$ → Calculate the errors of $\mathrm{E_{23}}$, $\mathrm{A_{23}}$.

IIb. $\mathbf{X} = [\mathbf{\Delta S_{23}^{\neq}},\ \mathbf{\Delta H_{23}^{\neq}}]\, m = 0 \cdot \mathbf{X}^{(0)} = [\mathbf{\Delta S_{23}^{\neq(0)}},\ \mathbf{\Delta H_{23}^{\neq(0)}}]$.

IIb1. Calculate $(\alpha_2(\mathrm{t_i})^{(0)})_{\mathrm{C}}$, $(\alpha_3(\mathrm{t}_i)^{(0)})_{\mathrm{C}}$(30).

(i) Calculate $T_i = \sum_{j=0}^{j=n} a_{j,i} t_i^j$

(ii) Calculate $k_{12}\,(T_i) = (k_B/h) \sum_{j=0}^{j=n} a_j t_i^j e^{-\Delta H_{12}^{\neq}/R \sum_{j=0}^{j=n} a_j t_i^j} e^{\Delta S_{12}^{\neq}/R}$

(iii) Calculate $k_{23}\,(T_i) = (k_B/h) \sum_{j=0}^{j=n} a_j t_i^j e^{-\Delta H_{23}^{\neq(0)}/R \sum_{j=0}^{j=n} a_j t_i^j} e^{\Delta S_{23}^{\neq(0)}/R}$

(iv) Numerical solution of the rate differential equations (20), (21).

IIb2. Calculate $\mathrm{SQD}^{(0)}$ (29).

IIb3. **AGDC Algorithm.**

IIb4. Optimized parameters: $\mathbf{X}^* = [\Delta \mathrm{S}_{23}^{\neq},\ \Delta \mathrm{H}_{23}^{\neq}]$ → Calculate the errors of $\Delta \mathrm{S}_{23}^{\neq}$, $\Delta \mathrm{H}_{23}^{\neq}$.

III. Statistical analysis of Residuals.

3 Computational aspects

The general computational treatment of ANN by means of the application of Matlab *"Neural Networks Toolbox"* with the creation of user's interfaces (GUI) including the appropriate analysis of Residuals and errors (MSE, SD, etc). In case of the numerical resolutions directly from the beginning of the ordinary differential equations (ODE), we have implemented mainly the followings Matlab functions: (a) ode45, based on an explicit Runge–Kutta formula, the Dormand–Prince pair. It is a one-step solver. In computing $y(t_n)$, it needs only the solution at the immediately preceding time point, $\mathrm{y(t_{n-1})}$. (b) ode23s is based on a modified Rosenbrock formula of order 2. Because it is a one-step solver, it can solve some kinds of *stiff* problems for which other are not effective; (c) ode23t is an implementation of the trapezoidal rule using a "free" interpolant (d) ode23tb is an implementation of TR-BDF2, an implicit Runge–Kutta formula with a first stage that is a trapezoidal rule step and a second stage that is a backward differentiation formula of order two and (e) ode15s is a multistep solver, variable order solver based on the numerical differentiation formulas (NDFs). Optionally, it uses the backward differentiation formulas (BDFs, also known as Gear's method. The numerical integrations have been performed using appropriate *quadrature*

formulas corresponding to a suitable numerical algorithms of resolution of integrals (Simpson, Lobatto, Gauss-Kronrod, Vectorized, etc.). With respect to computational application of AGDC algorithm, the program named KINNOISOT(AGDC), has been designed and performed in our laboratory by means of a computational executable codes (##.m type), in the Matlab environment using "M" language. The program is constituted by a Main program and several Functions or Subroutines.

4 Results and discussion

4.1 Analysis of the functions T/t and α/t

On the basis of the foregoing considerations, different types of functions T = f(t) have been used depending on the differential equation to be solved and/or the group of parameters to be determined. However, all of them have to fulfill the following requirements: (a) the functions must be monotonically increasing, in order to minimize the huge differences existing in the reaction rate of isothermal kinetics between the start and the end of the reaction (b) the increment in temperature produced in the interval of time values of non-isothermal kinetic data acquisition must be the appropriated one to achieve an extent of reaction of 75 % for α values of at least 2 *half-lives*. That leads to choose an appropriate rate of heating for each time value (c) the function must be accessible to experimentation and exactly reproducible in the laboratory. That is, the T values which are provided by the theoretical function T = f(t), have to be reached for each time value of the selected experimental interval.

We have selected the following functions:

1. function of polynomial type of n-th degree (15), without exact mathematical solution for the rate differential equations in which α_2 and α_3 are involved [(24), (25)] and Arrhenius and Eyring equations are considered;

2. inverse function corresponding to a hyperbolic branch (12) only for α_1 and Arrhenius equation,which permits to reach an exact mathematical solution for the rate differential equation.

The profiles of graphic plots of curves synthetically generated from inverse hyperbolic function show that as the value of "m" of Eq. (12) decreases a smoothing of the concavity of the hyperbolic branch is observed, tending towards linearity. The suitable value, and the one that satisfies the 3 requisites considered above, is $m = 4.20\,10^{-6}\,\mathrm{K}^{-1}\,\mathrm{min}^{-1}$. We fitted the T/t points to a first-degree polynomial function and then progressively to polynomials from 2nd to 5th degree, observing that the coefficients of the independent variable had negligible values with respect to the value of α_1 from the quadratic term. The values of the Statistical Analysis of Residuals show that the linear fitting can be taken as correct. This means that the T/t data pairs generated with the inverse hyperbolic function with a value of $m = 4.20\,10^{-6}\,\mathrm{K}^{-1}\,\mathrm{min}^{-1}$ can be satisfactorily fitted to a linear function. That is to say, the hyperbolic values of T can be emulated by means of a linear function and can be applied to the case of α_1 values when the Arrhenius equation is considered. At the laboratory, this allowed us to impose a linear heating rate (slope of $0.3734\,\mathrm{K\,min}^{-1}$), starting (t = 0) at T = 298.15 K.

The profiles of the non-isothermal kinetic curves for all species (α_1/t, α_2/t and α_3/t) sometimes show concave down segments while the isothermal curves are always

concave up. This can be explained in terms of the notion that in non-isothermal curves there are two opposing phenomena that affects the reaction rate: the increase in temperature with time increases the reaction rate, and the logical decrease in the concentration of reagents as they are consumed reduces it. Depending on which of the two phenomena predominates, one concavity or the other will be observed and even when both effects are balanced quasi-linear profiles appear.

4.2 Experimental design (ED)

The application of the computational method ANN demands to perform an appropriate previous organization of the experiments on the basis of the Experimental Design (ED). In the present work, we have chosen the most suitable one, the Central Star Composite Experimental Design (CSCED). The factors are formed by the set of parameters A_{12}/E_{12} and A_{23}/E_{23} for the Arrhenius equation and $\Delta H_{12}^{\neq}/\Delta S_{12}^{\neq}$ and $\Delta H_{23}^{\neq}/\Delta S_{23}^{\neq}$ when the Eyring equation is considered. The responses are the non-isothermal kinetic data of the base of the input curves (α_1/t, α_2/t, and α_3/t). It is necessary to consider 2 variables: (a) the extreme values of both factors in each of the 4 sets, which configure the experimental domain and (b) the relative values of each pair of ATP's parameters. Both variables must ensure that the binary combinations of the factors will generate a set of kinetic curves that will have sufficient information to ensure an optimal training process of the neural network. The number of levels of the factors of the ED must be suitable to avoid useless computational work and large differences in the spacing of the values of the responses. Accordingly, to optimize the ANN training process the kinetic curves of the input matrix must have efficient kinetic information and must be correctly distributed according to the choice of a suitable experimental design and an appropriate experimental domain. In agreement with the results of the study of the functions T = f(t), we have generated the non-isothermal kinetic data for both equations (Arrhenius and Eyring) in order to obtain the curves of the input matrix to perform the training process of the neural network from the ATPs organized according to the Experimental Design (ED) (Fig. 1).

4.3 Application of HA: results and discussion

The *training* process of the neural network is crucial in order to know its optimal architecture, which will guarantee the success of the ANN treatment. After testing a **large set** of different network topologies for the treatment of the kinetic system of consecutive reactions, we selected the one that provides the best results since this network architecture is the optimal architecture. The *training* processes, carried out for the 8 parameters by using the computation of the three kinetic curves (α_1 /t; α_2 /t and α_3 /t), have the following characteristics in common: *input matrices* formed by 45 non-isothermal kinetic curves with 50 pairs of kinetic data (α_1 /t) for the first reaction and 100 pairs (α_2 /t and α_3 /t) for the second one; value of the *random noise* of the order of the experimental error ($\pm 1.0\ 10^{-4}$); time interval to reach a 75 % extent of reaction; function T/t of hyperbolic branch type ($m = 4.2\ 10^{-6}\ K^{-1}min^{-1}$), only in the case

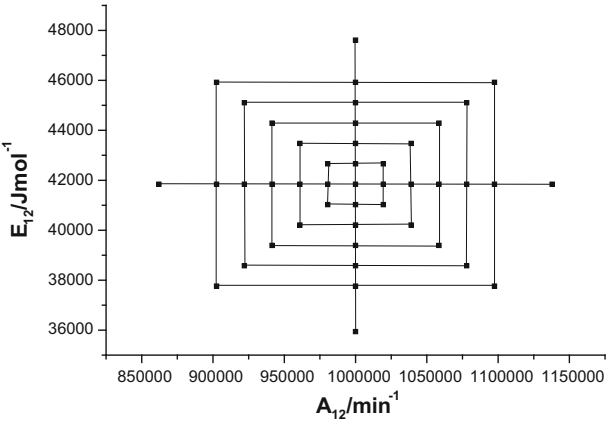

Fig. 1 Experimental Design (CSCED) constituted by 2 factors (A_{12}/E_{12}) and 45 points distributed in 9 levels (points 1–37) and 4 sub-levels: (points 38–45) taking part in the curve base of the *input matrix* used for the *training* process of the neural network

of the first reaction for the Arrhenius equation (A_{12} y E_{12}) and the polynomial type for both reactions and both equations (Arrhenius and Eyring) while the 6 remaining parameters are determined (A_{23}, E_{23}, $\Delta H_{12}^{\neq}$, $\Delta S_{12}^{\neq}$, $\Delta H_{23}^{\neq}$ and $\Delta S_{23}^{\neq}$).

We have carried out the *training* of the neuronal networks of the ANN treatment, using the same computational conditions for the 8 parameters grouped in pairs. They are the following ones: regarding to the *back-propagation* algorithm, we selected the Levenberg–Marquardt type for all the parameters because it provides the best optimized results for the *output* and *target* matrices; we have chosen 80/10/10 values for the percentages of computed data on the *training/validation/testing* processes, respectively; we considered a systematic variation of the number of *hidden layers* and their configurations (number of *nodes*) testing a complete set of possibilities for the topology of the neuronal networks in order to strictly determine their optimal architecture.

However we have confirmed the existence of a notorious ambiguity in the attained result that merits a detailed comment to avoid leading to wrong conclusions. Considering the graphic plots of the results of the *training, validation and testing ("ALL")* processes (Fig. 2) for a neuronal network with optimal structure (100,2,2) and configuration (14–10), we calculated $\Delta H_{23}^{\neq}$ and $\Delta S_{23}^{\neq}$ On this graphic, obtained from the simultaneous plot of both parameters, we determined a Regression line: *Output = 1.00xTarget + 1.30 10*$^{-6}$, with a value of $R = 0.99999$ that corresponds to a good linearity among the elements of the *output/target* matrices. Therefore, these results would be initially acceptable. However, some important considerations must be taken into account: the values of the parameters, which their order of magnitude is very different ($\Delta H_{23}^{\neq} \approx 10^4$ and $\Delta S_{23}^{\neq} \approx 10^2$), are represented on the same scale, so the points show up on the graphic, forming 2 groups separated by a long distance between them. In addition, all the corresponding values to $\Delta S_{23}^{\neq}$ are superimposed on the same point. This situation masks the real effects of dispersion, causing that the *output/target* Regression straight and the linear fitting parameters, obtained when both ATP parame-

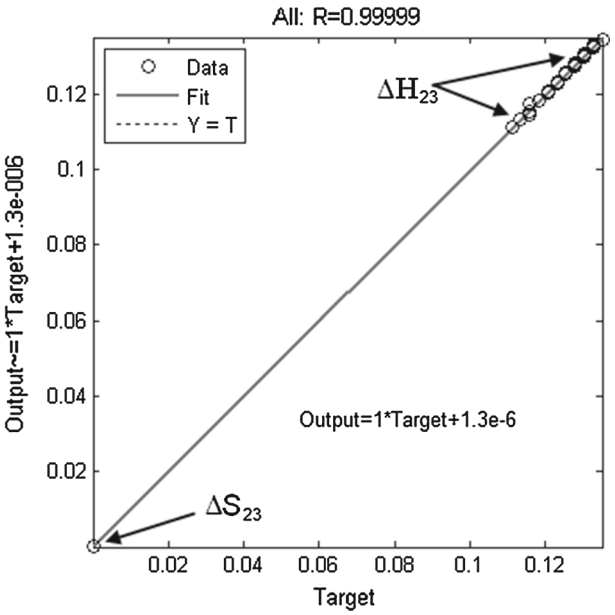

Fig. 2 Plot of the Regression line for the data from the elements of the *output versus target* matrices when both parameters ($\Delta H_{23}^{\neq}$ and $\Delta S_{23}^{\neq}$) are represented together for the processes of *training, validation* and *testing (ALL)* when ANN is applied to a neural network endowed with an architecture with 2 *hidden* layers (100, 2, 2)

ters are represented together on the graphic, are correct. However, when we consider the *output/target* Regression lines of each separate parameter by using the appropriate scale, a huge dispersion for the case of the $\Delta S_{23}^{\neq}$ parameter (Fig. 3, corresponding to the superimposed points observed on the Fig. 2) is detected. The obtained values are: 0.1285 for R^2, 0.3314 for the slope, (very far away from the value = 1) and 0.0383 for the ordinate at the origin (very different from 0). These values clearly manifest a very poor *output/target* correlation, what makes unacceptable the value of $\Delta S_{23}^{\neq}$ determined by ANN. Nevertheless, the results achieved in the $\Delta H_{23}^{\neq}$ case prove to have an acceptable linearity as it is shown on the Fig. 4.

In conclusion, these values are unacceptable, so it requires the complete application of HA to improve the results. That is, the obtained values for ATP's parameters after application of ANN method will be the *initial estimates* for the AGDC gradient algorithm, which will provide us very acceptable values for the final ATP's optimized parameters.

The values of the individual Errors and Standard Deviations of each of the 8 ATP's parameter ($SD(b_{ij})$) obtained after application of the ANN, with relative output and target values, can be determined from the following general expression for *SD*:

$$SD(b_{ij}) = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} \left[(b_{ij}^{Outputs} - b_{ij}^{Targets})/(b_{ij}^{Targets}) \right]^2}{n_p.n_c} \right)^{1/2} \quad (36)$$

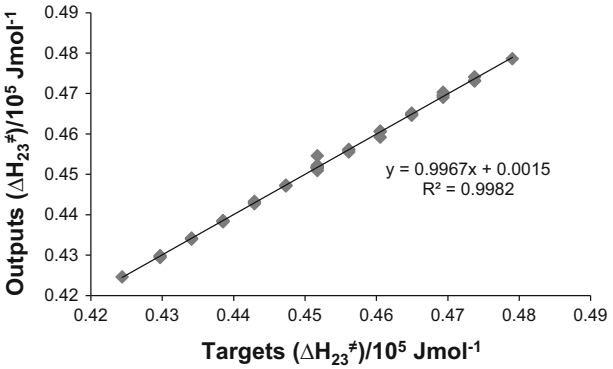

Fig. 3 Plot of the Regression line for the data from the elements of the *output versus target* values when $\Delta H_{23}^{\neq}$ is plotted separately for the process of *training, validation* and *testing (ALL)* of the ANN when the architecture of the neural network is constituted by 2 *hidden* layers (100, 2, 2).The equation obtained is: $Output = 0.9967 * Target + 0.0015 (R^2 = 0.9982)$

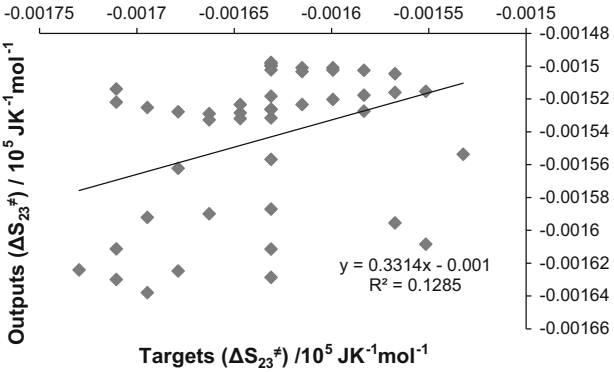

Fig. 4 Plot of the Regression line for the data from the elements of the *output versus target* values when $\Delta S_{23}^{\neq}$ is plotted separately for the process of *training, validation* and *testing (ALL)* of the ANN, when the architecture of the neural network is constituted by 2 *hidden* layers (100, 2, 2). The equation obtained is: $Output = 0.3314 * Target - 0.001 (R^2 = 0.1285)$

where b_{ij} are the ATP's parameters, n_c is the number of curves and n_p is the number of parameters per curve ($n_p = 1$ for the individual error calculation)

We carried out the *training* process of a large set of neuronal networks considering all the possible configurations for architectures with 2 and 3 *hidden* layers, discarding those of a single layer. In light of the high number of tested cases, we just only show the best results on the Tables 1 and 2. All of them were obtained from the most significant configurations of the curves corresponding to the *central point* of the 45 curves of the ED. The value pairs of *outputs*, A_{23}/E_{23} and $\Delta H_{23}^{\neq}/\Delta S_{23}^{\neq}$, and the deviations in % (Dev.%) respect to the values used to generate all the data, are showed on the first part of the Tables 1 and 2. In addition, the second part of the Tables shows the values of the optimized parameters after applying AGDC and the deviations in % (Dev.%) of the real values. After completing the whole application of HA we observed a great

Table 1 Results of the *training* process of neural networks constituted by architectures with 2 and 3 *hidden* layers and several configurations, when E_{23} and A_{23} are computed

ANN					ANN + AGDC			
Config	A_{23} (10^5 min^{-1})	E_{23} (10^4 J mol^{-1})	Dev.% (A_{23})	Dev.% (E_{23})	A_{23} (10^5 min^{-1})	E_{23} (10^4 J mol^{-1}	Dev.% (A_{23})	Dev.% (E_{23})
10/10	4.9626	4.2496	0.7466	0.0071	4.9740	4.2486	0.5207	0.0318
10/11	5.0512	4.2622	−1.0254	−0.2891	5.1192	4.2561	−2.3840	−0.1439
10/12	4.9752	4.3018	0.4959	−1.2200	5.0831	4.2543	−1.6619	−0.1003
10/14	4.9690	4.2858	0.6190	−0.8433	5.1641	4.2584	−3.2818	−0.1966
13/10	5.0672	4.2647	−1.3457	−0.3472	5.1469	4.2575	−2.9384	−0.1769
15/10	4.9809	4.2945	0.3814	−1.0470	5.1079	4.2356	−2.1979	−0.1326
20/10	5.0490	4.3250	−0.9813	−1.7665	5.0012	4.2501	−0.0235	−0.0016
7/10/10	5.0028	4.3079	−0.0576	−1.3625	5.0838	4.2543	−1.6762	−0.1017
9/10/10	5.1849	4.1884	−3.6988	1.4479	4.6510	4.2313	6.9801	0.4399

This table shows the obtained results of the ATP's parameters and their % deviations (Dev.%) when applying the algorithms ANN and HA (ANN+AGDC)

improvement just by comparing the values of Dev.%, (columns 8th and 9th) of both parameters with the ones calculated by the ANN application (columns 4th and 5th).

The improvement in the values of ATP's parameters is much better appreciated on a joint plot (Fig. 5) of the Residuals of α_2, calculated by using the values of parameters $\Delta H_{23}^{\neq}/\Delta S_{23}^{\neq}$, which were obtained from the ANN application (37), and the values determined by the HA algorithm application (38). In Fig. 5 we can appreciate this improvement, which maximum value is produced when a 40 % extent of reaction is reached by comparing the values of $Res(\alpha)_{HA} = 0.0013$ and $Res(\alpha)_{ANN} = 0.388$.

$$[Res(\alpha)_{ANN} = \alpha_{synt} - \alpha_{cal}(ANN)] \tag{37}$$

$$[Res(\alpha)_{HA} = \alpha_{synt} - \alpha_{cal}(HA)] \tag{38}$$

The *prediction* process, necessary to determine the final optimized parameters, consists in the application of HA to new kinetic curves, different from the considered ones at the *training* processes of neuronal networks. These new ones have been generated from ATP value pairs within the maximum intervals defined in the ED. The *prediction* constitutes the final process of the HA application to finally determine the unknown ATP parameters, corresponding to experimental kinetics carried out at the Laboratory. The obtained results are the value pairs of the parameters E_{23}/A_{23} (Table 3) and $\Delta H_{23}^{\neq}/\Delta S_{23}^{\neq}$ (Table 4). As for the *prediction* process, carried out after the ANN application, we used the optimal neuronal network after carrying out exhaustive processes of *training*, computing a base of 8 kinetic curves, the most suitable number to perform a multiple and simultaneous *prediction* of the ANN initial stage. After the AGDC algorithm application, we determine the final optimized values of the parameters, resulting from the complete application of the HA algorithm and the Dev.% between the Optimized Values and the ones used in the kinetic generation of data. After analyzing all

Table 2 Results of the *training* process for neural networks constituted by architectures with 2 and 3 *hidden* layers and several configurations, when $\Delta H_{23}^{\neq}$ and $\Delta S_{23}^{\neq}$ are computed

Config	ANN				ANN+AGDC			
	$\Delta S_{23}^{\neq}$ ($J\,K^{-1}\,mol^{-1}$	$\Delta H_{23}^{\neq}$ ($10^4\,J\,mol^{-1}$)	Dev.% ($\Delta S_{23}^{\neq}$)	Dev.% ($\Delta H_{23}^{\neq}$)	$\Delta S_{23}^{\neq}$ ($J\,K^{-1}\,mol^{-1}$)	$\Delta H_{23}^{\neq}$ ($10^4\,J\,mol^{-1}$)	Dev.% ($\Delta S_{23}^{\neq}$)	Dev.% ($\Delta H_{23}^{\neq}$)
8/10	−163.7169	4.5016	−0.3782	0.3451	−163.6531	4.5000	−0.3391	0.3811
10/9	−164.0223	4.4914	−0.5655	0.5704	−163.9694	4.4902	−0.5334	0.5972
10/11	−160.5622	4.4979	1.5559	0.4251	−162.2034	4.5449	0.5498	−0.6142
10/12	−179.0484	4.4927	−9.7783	0.5410	−164.6662	4.4685	−0.9603	1.0782
10/15	−136.3193	4.5143	16.4197	0.0625	−158.9482	4.6457	2.5455	−2.8456
10/20	−178.3090	4.5212	−9.3250	−0.0898	−162.5956	4.5329	0.3093	−0.3482
13/10	−160.4780	4.5002	1.6075	0.3752	−162.1255	4.5774	0.5975	−0.6679
14/10	−163.7297	4.5430	−0.3861	−0.5714	−163.0215	4.5196	0.0481	−0.0531
15/10	−161.7405	4.5239	0.8335	−0.1499	−162.3150	4.5416	0.4813	−0.5405
9/10/10	−161.9950	4.5061	0.6774	0.4363	−162.7351	4.5285	0.2237	−0.2508
10/9/10	−181.5295	4.5271	−11.2995	0.5704	−159.9146	4.6161	1.9530	−2.1899
10/10/10	−162.6242	4.5210	0.2916	0.1582	−162.7918	4.5268	−0.1889	−0.2125
10/10/13	−161.8261	4.4894	0.7810	−0.0898	−162.9366	4.5222	0.1002	−0.1109
10/10/14	−160.3689	4.5145	1.6744	−0.7670	−161.8253	4.5567	0.7815	−0.8754
10/12/10	−163.6233	4.5064	−0.3208	−0.1499	−163.5313	4.5038	−0.2644	0.2973
10/15/10	−165.0704	4.5121	−1.2081	0.1113	−164.1942	4.4830	−0.6709	0.7566
10/20/10	−182.9596	4.5263	−12.1763	−0.2026	−161.2712	4.5745	1.1213	−1.2682

This table shows the obtained results of the ATP's parameters and their % deviations (Dev.%) when applying the algorithms ANN and HA (ANN+AGDC)

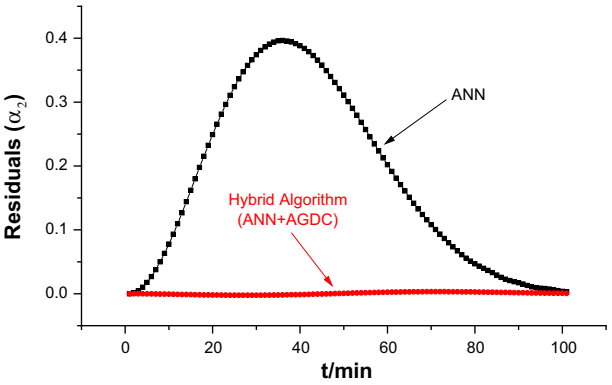

Fig. 5 Plot of the *Residuals* of α_2 using the values of the ATP's parameters ($\Delta H_{23}^{\neq}$ and $\Delta S_{23}^{\neq}$) obtained by applying ANN method [Eq. (37)] and those obtained from Eq. (38) after the application of the complete algorithm HA

Table 3 Results of the 4 *prediction* processes in the case of the optimal architecture of neural network (100, 2, 2) when Arrhenius equation is considered and A_{23}/E_{23} values are predicted, after application of HA (ANN+AGDC)

HA Algorithm (ANN + AGDC)					
Real values (A_{23}) (10^5 min^{-1})	Real values (E_{23}) (10^4 J mol^{-1})	Optimized values (A_{23}) (10^5 min^{-1})	Optimized values (E_{23}) (10^4 J mol^{-1})	Dev.% (A_{23})	Dev.% (E_{23})
5.0830	4.0306	5.7654	5.3756	30.0620	−17.3260
4.9202	4.1031	3.6072	4.1655	27.8558	1.9876
5.0173	4.0928	5.0621	4.0925	−3.2565	5.5473
4.9789	4.2181	5.0090	3.9874	5.3773	0.3499

Real and optimized values can be compared after evaluation of the values of their deviations (Dev.%)

Table 4 Results of the 8 *prediction* processes in the case of the optimal architecture of neural network (100, 2, 2) when Eyring equation is considered and $\Delta H_{23}^{\neq}/\Delta S_{23}^{\neq}$ values are predicted, after application of HA (ANN+AGDC)

HA algorithm (ANN+ AGDC)					
Real values ($\Delta S_{23}^{\neq}$) (J K^{-1} mol^{-1})	Real values ($\Delta H_{23}^{\neq}$) (10^4 J mol^{-1})	Optimized values ($\Delta S_{23}^{\neq}$) (J K^{-1} mol^{-1})	Optimized values ($\Delta H_{23}^{\neq}$) (10^4 J mol^{-1})	Dev.% ($\Delta S_{23}^{\neq}$)	Dev.% ($\Delta H_{23}^{\neq}$)
−159.5642	4.3408	−158.1711	4.2976	−0.9136	0.9999
−163.8140	4.7031	−166.6814	4.7811	1.6402	−1.8692
−152.7474	4.3832	−155.5510	4.4701	1.7222	−1.9387
−150.1146	4.4330	−156.0480	4.5482	2.4209	−2.6887
−159.0111	4.6072	−163.3297	4.6994	1.7745	−2.0446
−150.1152	4.4734	−158.5474	4.5650	1.8465	−2.0531
−153.2067	4.5551	−159.5911	4.7207	3.0962	−3.4960
−152.6486	4.5172	−1579356	4.6775	3.1664	−3.5481

Real and optimized values can be compared after evaluation of the values of their deviations (Dev.%)

Table 5 Final values of the 8 ATP's parameters (A_{12}, E_{12}, A_{23}, E_{23}, $\Delta H^{\neq}_{12}$, $\Delta S^{\neq}_{12}$, $\Delta H^{\neq}_{23}$ and $\Delta S^{\neq}_{23}$) obtained after application of the HA and their optimal architecture and configurations *(nodes)* of the neural networks used in the ANN treatment

ATP	Optimun architecture	Nodes	HA final values
A_{12}	(50,3,2)	10/20/10	1.10^6 (*1*)
E_{12}	(50,3,2)	10/20/10	41840 (*2*)
$\Delta H^{\neq}_{12}$	(50,3,2)	10/9/10	38285 (*3*)
$\Delta S^{\neq}_{12}$	(50,3,2)	10/9/10	−172.7 (*4*)
A_{23}	(100,2,2)	20/10	5.10^5 (*1*)
E_{23}	(100,2,2)	20/10	42500 (*2*)
$\Delta H^{\neq}_{23}$	(100,2,2)	14/10	45172 (*3*)
$\Delta S^{\neq}_{23}$	(100,2,2)	14/10	−163.1 (*4*)

(*1*) min^{-1}, (*2*) $J\,mol^{-1}$, (*3*) $J\,mol^{-1}$, (*4*) $J\,mol^{-1}K^{-1}$

the values of the *Statistic moments* of errors $[SD(b_{ij})]$ and Dev.%, according to Eq. (37), and (38), we can easily know the optimal architecture in terms of structure and configuration of the *hidden* layers used to determine the 8 ATP parameters and their values (Table 5).

5 Conclusions

The obtained results of the HA application demonstrate its great robustness when it is applied to the calculation of the ATP's parameters of the kinetic system of the first order consecutive reactions. Additionally, the HA embodies a satisfying versatility, providing evidences of its applicability to other models and kinetic systems in order to obtain other kind of kinetic parameters. The great advantage of HA is the efficient resolution of the ambiguity of the results determined by ANN: not only discriminates between the pairs of parameters obtained in the *output* matrices, but also calculates their individual values with considerable accuracy and precision. Furthermore, the exhaustive control and corrections performed over the movement vector, solves other weak point of the classic optimization algorithms avoiding the possibility of being trapped at local minima, saddle points...etc. This allows to AH successfully determine the values of the 8 ATP's parameters involved in the studied kinetic system (A_{12}, E_{12}, A_{23}, E_{23}, $\Delta H^{\neq}_{12}$, $\Delta S^{\neq}_{12}$, $\Delta H^{\neq}_{23}$ and $\Delta S^{\neq}_{23}$), of which high-number of parameters and very different orders of magnitude make the task extremely complicated for the classic algorithms of mathematical optimization.

The lack of necessity to know a priori the order of magnitude and the sign of the 8 parameters represents another important advantage. Therefore, the initial ANN treatment provides the values of the all ATP's parameters used by AGDC as *initial estimates*. In addition, the computational application, in which the HA is implemented, doesn't present difficulties to non-expert users.

Finally, in comparison with the isothermal kinetics, the evaluation of non-isothermal kinetics produces evident advantages in the experimental aspect, generating significant savings both in reagents and laboratory time. This allows one to perform only one replicated non-isothermal kinetic since it is sufficient for the computational treatment with the HA proposed in this paper.

References

1. A.A. Frost, R.G. Pearson, *Kinetics and Mechanism* (Wiley, New York, 1961)
2. C.H. Bandford, C.F.H. Tipper, *Comprehensive Chemical Kinetics* (Elsevier, Amsterdam, 1973)
3. M.A. Wolfe, *Numerical Methods for Unconstrained Optimization* (Van Nostrand, New York, 1978)
4. S. Vadja, H. Rabitz, J. Phys. Chem. **92**, 701–707 (1988)
5. S. Vadja, H. Rabitz, J. Phys. Chem. **98**, 5265–5271 (1994)
6. J. Zupan, J. Gasteiger, *Neural Networks for Chemists* (VCH, Weinheim, 1993)
7. M.M. Canedo, J.L. González-Hernández, Chemom. Intell. Lab. Syst. **66**, 63–78 (2003)
8. M.M. Canedo, J.L. González-Hernández, Chemom. Intell. Lab. Syst. **66**, 93–97 (2003)
9. M.M. Canedo, J.L. González-Hernández, J. Math. Chem. **49**, 163–184 (2011)
10. M.M. Canedo, J.L. González-Hernández, S. Encinar, Appl. Math. Comput. **219**, 7089–7101 (2013)
11. J.L. González-Hernández, M.M. Canedo, C. Grande Martin, MATCH Commun. Comput. Chem. **70**, 951–970 (2013)
12. A. Niazi, R. Leardi, J. Chemom. **26**, 345–351 (2012)
13. M. Maeder, Y.M. Neuhold, G. Puxty, Chemom. Intell. Lab. Syst. **70**, 193–203 (2004)
14. C. Hervás, J.A. Algar, M. Silva, J. Chem. Inf. Comput. Sci. **40**, 724–731 (2000)
15. S. Curteanu, H. Cartwright, J. Chemom. **25**, 527–549 (2011)
16. F. Amato, J.L. González-Hernández, J. Havel, Talanta **93**, 72–78 (2012)
17. S. Encinar, J.L. González-Hernández, M. Mar Canedo, J. Math. Chem. **51**, 1634–1653 (2013)
18. S. Encinar, J.L. González-Hernández, M. Mar Canedo, MATCH Commun. Comput. Chem. **72**, 427–450 (2014)
19. K.J. Laidler, Pure Appl. Chem. **68**, 149–192 (1996)
20. T. Kohonen, Neural Networks. **1**, 3–16 (1988)
21. MathWorks MatLab, R2012a, Vs7.14.0.739 (2012)
22. P. Gill, W. Murray, M.H. Wright, *Practical Optimization* (Academic Press, London, 1981)
23. C.F. Gerald, P.O. Wheatley, *Applied Numerical Analysis* (Adinson-Wesley, Massachusetts, 1984)
24. J. Topping, *Errors of Observation and Their Treatment* (Chapman and Hall, London, 1978)
25. R.D. Cook, S. Weisberg, *Residuals and Influence in Regression* (Chapman and Hall, New York, 1982)