Journal of Chromatography A, 942 (2002) 93–105

JOURNAL OF
CHROMATOGRAPHY A

www.elsevier.com/locate/chroma

On the use of genetic algorithms for response surface modeling in high-performance liquid chromatography and their combination with the Microsoft Solver

P. Nikitas*, A. Pappa-Louisi, A. Papageorgiou, A. Zitrou

Laboratory of Physical Chemistry, Department of Chemistry, Aristotle University of Thessaloniki, 54006 Thessaloniki, Greece

Received 15 May 2001; received in revised form 2 October 2001; accepted 2 October 2001

Abstract

Four genetic algorithms — the classical, Haupt's, Brunetti's and a modification of the classical algorithm suggested in the present paper — are examined when they are used for the modeling of response surfaces in high-performance liquid chromatography (HPLC). We found that the best results are obtained from our modification and the worst by Haupt's algorithm. The classical genetic algorithm gives satisfactory results, better than those of Brunetti's algorithm. We also ascertained that all genetic algorithms may get stuck in a local minimum other than the global one, except for our modification, which can be considered to approach a global method. Finally, the time needed for the optimization of a genetic algorithm and the combination of a genetic algorithm with a non-linear least-squares routine are considered and discussed. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Genetic algorithms; Response surface modeling; Fitting procedures; Modeling

1. Introduction

In a recent paper the modeling of high-performance liquid chromatography (HPLC) response surfaces has been studied by means of a new technique which combines a non-linear curve-fitting procedure with a Monte Carlo search for initial estimates [1]. This technique has been applied to experimental systems and we found that it is, in general, a global method, i.e. a method that can find the global minimum, whereas a non-linear least-squares technique alone is likely to be trapped into a local minimum other than the global one and/or to pseudo-solutions [1]. A pseudo-solution is a solution obtained by a non-linear least-squares routine that does not correspond to a local minimum but it arises from the fact that at least one of the fitting parameters does not affect the value of the sum of squares of residuals (χ^2) when this parameter varies within a certain range of values. For this reason each pseudo-solution appears just one time when we use the Monte Carlo routine described in [1].

It is known from literature that the genetic algorithms offer an alternative solution to the above problem, since they may determine the global minimum of a function without being trapped into local minima or pseudo-solutions [2–5]. Despite this interesting feature, the application of genetic algo-

*Corresponding author. Tel.: +30-31-997-773; fax: +30-31-997-709.

E-mail address: nikitas@chem.auth.gr (P. Nikitas).

PII: S0021-9673(01)01353-X

rithms to HPLC problems is very limited. Thus, according to our knowledge, there are two papers appearing in 1993 concerning strictly the use of genetic algorithms for the modeling of response surfaces in HPLC [5,6]. However, since then significant progress has been made in genetic algorithms: The replacement of the binary or Gray coding by real numbers, the development of various types of crossover, the use of non-linear mutation [3,4] and recently the development by Brunetti of a novel algorithm suitable for non-linear least-squares procedures [7].

There are two main problems concerning the use of genetic algorithms: first, there is not just one genetic algorithm and therefore we have to choose which is the most effective; second, they require the appropriate adjustment of a number of parameters, such as crossover probability, mutation probability, type of mutation, size of population etc., a procedure that is time consuming. In the present paper we used the classical algorithm, the algorithms proposed by Haupt and Brunetti and a modification of the classical genetic algorithm proposed here. We checked which of the above algorithms is more effective and whether by adjusting the parameters of a genetic algorithm in a certain experimental system we can use these parameters for other systems as well. Finally, the possibility of combining a genetic algorithm with a non-linear least-squares procedure is also discussed.

2. On genetic algorithms

2.1. Classical genetic algorithm

Genetic algorithms are procedures that imitate the natural evolution process and can be used for the computation of the global maximum or minimum of a function [2–4]. The basic scheme is the following: *creation of original population→evaluation of cost→selection of mate→reproduction→mutation→ new population*. In the new population (generation) the same procedures are applied, i.e. evaluation of cost, mate selection, reproduction, mutation, and the whole circle is repeated m ($m>1000$) times.

The population consists of N *chromosomes*, that is, arrays of q real numbers ($x_1, x_2, \dots, x_q$), where q is the number of adjustable parameters, which the global minimum depends on. The members of a chromosome are the *genes* $x_1, x_2, \dots, x_q$, and they are the adjustable parameters that need to be determined. The genes are selected by means of random numbers from appropriate regions where they are supposed to exist. Therefore, the initial population is randomly generated within pre-established ranges of the genes $x_1, x_2, \dots, x_q$, by using a random seed which is different in each run of the genetic algorithm. Note that this algorithm as well as all the genetic algorithms we used are of continuous parameter, that is, the genes are real and not binary numbers [2–4].

In least-squares fitting the target is to determine the global minimum of the function:

$$\chi^2 = \sum_{i=1}^{n} (y_{\mathrm{i,exp}} - y_{\mathrm{i,calc}})^2 \qquad (1)$$

where $y_{\mathrm{i,exp}}$ are the experimental data and $y_{\mathrm{i,calc}}$ are the corresponding calculated values from the function to be fitted. In our case, this function describes retention surfaces in HPLC, i.e. surfaces concerning the combined effect of pH and modifier concentration on the capacity factor k of a solute. The adjustable parameters of the above function are the genes $x_{1,}$ $x_2, \dots$ and the calculation of the value of χ^2 at each chromosome is called *evaluation cost*.

Mate selection is the operation where couples of chromosomes are selected from the initial population. There are several ways to mate the chromosomes. The main procedure is by means of the roulette wheel [2–4]. In order to increase the performance of this operation we used scaling, according to Goldberg's suggestion [2–4]. The mate selection procedure creates couples of the general type: $x_{\mathrm{male}}=(x_{1\mathrm{m}}, x_{2\mathrm{m}}, \dots.., x_{\mathrm{qm}})$, $x_{\mathrm{female}}=(x_{1\mathrm{f}}, x_{2\mathrm{f}}, \dots.,$ $x_{\mathrm{qf}})$. Not all the members of the population can mate. The percentage of the population chromosomes that mates is called *crossover probability*.

Reproduction is achieved by the *crossover* operation, which is a partial exchange of genetic content. There are various types of crossover [2–4]. Here, we adopted the combined crossover technique [2–4]. In more detail, a gene, say x_4, is selected by means of random numbers and the following offspring are produced:

$$x_{\text{child1}} = (x_{1f}, x_{2f}, x_{3f}, v_{4m}, x_{5m}, x_{6m}, \ldots\ x_{qm})$$

$$x_{\text{child2}} = (x_{1m}, x_{2m}, x_{3m}, v_{4f}, x_{5f}, x_{6f}, \ldots\ x_{qf})$$

where $v_{4m} = \beta x_{4m} + (1-\beta)x_{4f}$, $v_{4f} = \beta x_{4f} + (1-\beta)x_{4m}$ and β is a random number in the region [0,1]. Offspring replace their parents.

Mutation alters one gene of a selected chromosome by a random change with a probability equal to the *mutation rate*. In the present paper, uniform or Gaussian mutation has been used [3].

The operations of *mate selection*, *reproduction* and *mutation* create the *new population* from the original one and the whole circle is repeated several times. The performance of the algorithm is enhanced by finding the best chromosome of one generation, i.e. the chromosome that gives in fitting procedures the minimum value of the cost function, and transferring it to the next generation.

2.2. Haupt's algorithm [4]

In this genetic algorithm the members of a generation (chromosomes) are sorted according to their cost value, i.e. the value of χ^2, from low to high values. Then the population is divided in two groups: the first N_{good} chromosomes comprise the good chromosomes, whereas the rest $N_{\text{bad}} = N - N_{\text{good}}$, with the relatively high values of χ^2, are considered to be the bad chromosomes. Mating takes place among the N_{good} chromosomes and offspring replace the N_{bad} chromosomes of the population. Mutation is applied to the chromosomes of the total population. After mutation the chromosomes are sorted again and the circle is repeated until a proper stop criterion is fulfilled.

2.3. Brunetti's algorithm [7]

The essence of Brunetti's algorithm is the lack of mutations. Due to this feature, the algorithm converges rapidly to a local or global minimum. The lack of mutations also results rapidly in the decrease of the variance of the cost values of the population. When the variance value falls below a certain limit (variance edge value = 0.1 or 0.01 or 0.001), a new population is created from the beginning with $N-1$ members. The best chromosome of the previous population is selected to be the N^{th} member of the new population and the whole procedure is repeated until a stop criterion is fulfilled.

2.4. Our modification

We noticed that a genetic algorithm in certain systems may converge to a local or global minimum after a relatively small number of generations, 2000–5000. In these systems no significant improvement is made by increasing the number of generations, especially when we use Gaussian mutation. Therefore, an apparent modification of the classical algorithm would be the following: we run the classical algorithm for *m* generations, *m* ranging from about 2000 to 5000. Then, an entirely new population is generated with $N-1$ members (chromosomes) and the best chromosome of the previous population is added as the N^{th} member. This procedure is repeated *M* times and the best chromosome of all these populations is the final solution.

The above modification of the classical genetic algorithm may be also considered as a modification of Brunetti's algorithm on the following two points: (a) the algorithm we propose includes the creation of mutations at each generation, and (b) in Brunetti's algorithm the new population is created when the variance value falls below a certain limit, whereas in our algorithm this happens when the number of generations reaches a certain value.

3. Application to the modeling of HPLC retention surfaces

In the present paper we used the above algorithms to model retention surfaces in HPLC by determining the coefficients of the theoretical equation adopted to describe these surfaces. In particular, five retention data sets were fitted to a selected theoretical equation by means of the genetic algorithm we examined. Four of these retention data sets, concerning the capacity factors of adenosine, hydroxyzine and benzoic acid in buffered mobile phases of different pH modified with methanol as well as the retention data of adenosine in mobile phases modified with acetonitrile, were taken from literature [1,5,8,9]. The fifth data set, concerning the capacity factors of 5-hydroxyindole-3-acetic acid in isopropanol buffers, was studied experimentally in this work.

The equations adopted for the theoretical description of the experimental data were the following [10,11]:

$$k=\frac{k_0^0 K_\mathrm{b}^0\,\mathrm{e}^{(q_1+s_0)\varphi+(q_2+t_0)\varphi^2}+k_1^0 10^{-\mathrm{pH}}\,\mathrm{e}^{s_1\varphi+t_1\varphi^2}}{K_b^0\,\mathrm{e}^{q_1\varphi+q_2\varphi^2}+10^{-\mathrm{pH}}} \tag{2}$$

for a weak monoprotic base, and

$$k=\frac{k_0^0\,\mathrm{e}^{s_0\varphi+t_0\varphi^2}+k_1^0 K_\mathrm{a}^0\,10^{\mathrm{pH}}\,\mathrm{e}^{(s_1+q_1)\varphi+(t_1+q_2)\varphi^2}}{1+K_\mathrm{a}^0\,\mathrm{e}^{q_1\varphi+q_2\varphi^2}\,10^{\mathrm{pH}}} \tag{3}$$

for a weak monoprotic acid. Here, φ is the percentage or the volume fraction of an organic modifier in a mobile phase of a given pH, k_0^0 and k_1^0 are capacity factors of the neutral and charged species, respectively, extrapolated to a pure aqueous buffer, s_0 and t_0 are parameters describing the variation of k_0 with φ ($\ln k_0=\ln k_0^0+s_0\varphi+t_0\varphi^2$), s_1 and t_1 are the corresponding parameters of k_1 ($\ln k_1=\ln k_1^0+s_1\varphi+t_1\varphi^2$), K_b^0 and K_a^0 are the dissociation constants of the equilibria $\mathrm{BH}^+\Leftrightarrow\mathrm{B}+\mathrm{H}^+$ and $\mathrm{HA}\Leftrightarrow\mathrm{H}^++\mathrm{A}^-$, respectively, extrapolated to a pure aqueous mobile phase, and q_1 and q_2 are parameters describing the variation of K_a or K_b ($\ln K=\ln K^0+q_1\varphi+q_2\varphi^2$). Note that from the solutes we studied adenosine behaves like a monoprotic base, whereas all the other solutes exhibit the behavior of monoprotic acids.

As concerns the χ^2 function (Eq. (1)), it acquires the form:

$$\chi^2=\sum_{i=1}^{n}(k_{\mathrm{i,exp}}-k_{\mathrm{i,calc}})^2 \tag{4}$$

where $k_{\mathrm{i,exp}}$ are the experimental values of the capacity factor and $k_{\mathrm{i,calc}}$ are the corresponding values calculated from either Eqs. (2) or (3).

4. Experimental

As noted above, the capacity factors of 5-hydroxyindole-3-acetic acid in isopropanol buffers were studied experimentally in this work. The main experimental conditions are the following. The experiments were carried out on a Shimadzu LC-9A HPLC system using a Gilson electrochemical (EC) detector set at 0.6 V vs. the Ag/AgCl reference electrode. A 250×4 mm (5 μm inertsil ODS-3) MZ$^-$ analytical column was used. The column was thermostatted at 30°C. Different mobile phases consisting of an aqueous phosphate buffer [12] and isopropanol were used. The total ionic strength of the mobile phases was held constant at 0.02 *M*. Isopropanol concentrations of 0, 0.8, 2, 3 and 5% were used. At each composition six different pH values were studied in the pH range 3.1 to 7.74, as given in Table 1. The concomitant effects of pH and isopropanol concentration were studied on the elution behavior of 5-hydroxyindole-3-acetic acid. The flow-rate was set at 1.0 ml/min. The hold-up, t_0, was estimated as 1.88 min. A volume of 20 μl of an aqueous standard solution containing 5 μg/ml of 5-hydroxyindole-3-acetic acid was injected. More experimental details

Table 1
Retention times of 5-hydroxyindole-3-acetic acid in isopropanol–aqueous buffer mobile phase as a function of pH and the volume fraction of isopropanol, φ

pH	φ	*t*, min	pH	φ	*t*, min	pH	φ	*t*, min
3.10	0.00	211.0	4.52	0.00	102.00	6.89	0.00	21.90
3.10	0.008	81.6	4.52	0.008	40.90	6.89	0.008	11.20
3.10	0.02	38.0	4.52	0.02	19.50	6.89	0.02	6.64
3.10	0.03	24.6	4.52	0.03	14.10	6.89	0.03	5.26
3.10	0.05	14.3	4.52	0.05	8.37	6.89	0.05	3.77
3.54	0.00	210.0	5.25	0.00	58.20	7.74	0.00	18.50
3.54	0.008	80.7	5.25	0.008	25.40	7.74	0.008	9.83
3.54	0.02	36.6	5.25	0.02	13.20	7.74	0.02	6.18
3.54	0.03	24.7	5.25	0.03	9.44	7.74	0.03	4.90
3.54	0.05	13.9	5.25	0.05	6.00	7.74	0.05	3.61

were described in our previous works, see for example Ref. [8].

The genetic algorithms were written in Visual C++ and the calculations were carried out on a PC with a 600 MHz Pentium III processor. The performance of the genetic algorithms used in this paper was tested in comparison with the performance of the genetic algorithm suggested by Michalewicz [3] using many of the functions suggested in [3]. We found that, depending on the function, the performance of the algorithms used in this paper is comparable and better than that of Michalewicz. It should be noted that for the evaluation of the performance of a genetic algorithm we may adopt three criteria: (a) The ability of a genetic algorithm to locate the global optimum, (b) the time needed for that, and (c) the reproducibility, i.e. the probability that a global optimum is found when a genetic algorithm is run once. The reproducibility is a crucial criterion for the performance of a genetic algorithm since if, for example, it is very small, say less than 1%, the algorithm in fact cannot locate the global optimum and therefore its performance is bad. In what concerns the time criterion, in the present study the operation time of a genetic algorithm is determined by the maximum number of generations and therefore this criterion cannot be used for the performance of a genetic algorithm.

5. Results and discussion

We first model the response surfaces of the above experimental systems using the Monte Carlo method described in [1]. Some of the results obtained are reported in Table 2. In particular, Table 2 shows the adjustable parameters that correspond to the global minimum of χ^2 and the probability with which this minimum is found by the Monte Carlo. The regions where the adjustable parameters were searched are shown in Table 3. The standard errors of the adjustable parameters of Table 2, calculated by the curvature matrix method using the CM1 procedure [1], are listed in Table 4. It is seen that the uncertainty in the calculated values of the model coefficients q_2, t_0 and t_1 is high enough, indicating that no chromatographic insights can be gained from the values of these parameters. Note that, according to the theory [13,14] the values of t_0 and t_1 must be positive numbers, but the high uncertainty cannot verify this prediction. It should be also noted that for the system of hydroxyzine in methanol–aqueous buffer mobile phase we have chosen $q_1 = q_2 = 0$, because (a) the experimental data are relatively few and for this reason we reduced the number of the adjustable parameters to avoid over-fitting complications, and (b) the treatment of q_1, q_2 as adjustable parameters do not improve considerable the fitting

Table 2
Values of (a) the adjustable parameters of Eqs. (2) or (3), (b) χ^2 and (c) the percentage of the successful attempts in finding the global minimum

System[a]	1	2	3	3[b]	4	5
k_0^0	22.566	22.349	21.252	17.725	45.974	119.712
K_a^0, K_b^0	3.33×10^{-4}	3.23×10^{-4}	4.81×10^{-6}	4.81×10^{-6}	5.09×10^{-5}	4.14×10^{-5}
q_1	22.838	−9.512	0	0	0	−43.141
s_0	−79.854	−24.212	−19.949	−11.619	3.914	−140.969
q_2	−1366.00	105.609	0	0	0	2845.059
t_0	706.816	79.823	4.100	4.100	−11.336	2472.737
k_1^0	0.925	1.166	76490.7	7.100	0.168	10.959
s_1	−9.816	1.956	7.472	−8.030	−7.328	−91.991
t_1	41.681	−99.371	−11.925	−11.924	0.609	1199.669
χ^2	3.498	7.824	1.068	1.068	0.127	131.37
%	25	96	16	97	66	34

[a] Systems: 1 = adenosine in acetonitrile–aqueous buffer; 2 = adenosine; 3 = hydroxyzine; 4 = benzoic acid in methanol–aqueous buffer; 5 = 5-hydroxyindole-3-acetic acid in isopropanol–aqueous buffer mobile phase.

[b] Fitted to the modification of Eq. (3) discussed in the text.

Table 3
Ranges of the adjustable parameters of Eqs. (2) or (3) for the experimental systems studied

System[a]	1		2		3		3[b]		4		5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
k_0^0	0	50	0	50	0	100	0	100	0	10	50	200
K_a^0,K_b^0	0	0.1	0	0.1	0	0.01	0	0.01	0	0.01	0	0.01
q_1	−1000	1000	−1000	1000	0	0	0	0	0	0	−300	300
s_0	−1000	1000	−1000	1000	−100	100	−100	100	−100	100	−300	300
q_2	−10000	10000	−10000	10000	0	0	0	0	0	0	−5000	5000
t_0	−10000	10000	−10000	10000	−100	100	−100	100	−100	100	−500	5000
k_1^0	0	10	0	10	50000	100000	0	100	0	200	0	20
s_1	−1000	1000	−1000	1000	−100	100	−100	100	−100	100	−500	500
t_1	−10000	10000	−10000	10000	−100	100	−100	100	−100	100	−2000	2000

[a] Systems as in Table 2.

[b] Ranges of the adjustable parameters of the modified Eq. (3) for hydroxyzine in methanol–aqueous buffer mobile phase.

($\chi^2=1.001$ instead of 1.068). The same choice of $q_1=q_2=0$ has been adopted for benzoic acid in methanol–aqueous buffer mobile phase.

The most interesting result of Table 2 concerns the system of hydroxyzine in methanol–aqueous buffer mobile phase. We observe an extremely high value of $k_1^0=76490.7$ much higher than the value of $k_0^0=21.252$. However, this is physically meaningless, since for an ionogenic solute the capacity factor of charged species (k_1^0) should be less than the neutral one (k_0^0). This peculiar behavior has its origin to the fact that k_1^0 is the extrapolated value of k_1 via the equation $\ln k_1 = \ln k_1^0 + s_1\varphi + t_1\varphi^2$ and similarly k_0^0 is the extrapolated value of k_0 via the equation $\ln k_0 = \ln k_0^0 + s_0\varphi + t_0\varphi^2$. However, for the system of hydroxyzine in methanol–aqueous buffer mobile phases we use three values of $\varphi=0.65$, 0.70, 0.75 and therefore the extrapolation to $\varphi=0$ is a completely untrustworthy and eventually erroneous procedure. This problem can be overcome if we modify Eq. (3) and use instead of $\ln k_0 = \ln k_0^0 + s_0\varphi + t_0\varphi^2$ and $\ln k_1 = \ln k_1^0 + s_1\varphi + t_1\varphi^2$ the following equations:

$$\ln k_0 = \ln k_{0,\mathrm{r}}^0 + s_0(\varphi - \varphi_\mathrm{r}) + t_0(\varphi - \varphi_\mathrm{r})^2 \text{ and } \ln k_1 = \ln k_{1,\mathrm{r}}^0 + s_1(\varphi - \varphi_\mathrm{r}) + t_1(\varphi - \varphi_\mathrm{r})^2 \tag{5}$$

where φ_r is a reference value of φ. In our case we have selected $\varphi_\mathrm{r}=0.65$ and therefore $k_{0,\mathrm{r}}^0$, $k_{1,\mathrm{r}}^0$ are referred to this value of φ. An analogous modification of Eq. (3) has been suggested by Marques et al. [5]. The results obtained using the above modification of Eq. (3) are shown in Table 2 in column 3[b]. Note that the global minimum is not affected by the use of Eqs. (5). Note also the great increase in the performance of the Monte Carlo method when

Table 4
Standard errors of the adjustable parameters of Table 2

System[a]	1	2	3	3[b]	4	5
k_0^0	0.28	0.38	190.94	0.50	9.39	2.42
K_a^0, K_b^0	4.36×10^{-5}	6.35×10^{-5}	1.02×10^{-6}	1.02×10^{-6}	1.9×10^{-6}	4.8×10^{-5}
q_1	21.95	13.85	–	–	–	38.27
s_0	3.20	1.18	67.66	4.24	24.77	8.59
q_2	401.27	118.47	–	–	–	876.01
t_0	91.27	10.33	48.96	49.04	32.41	389.45
k_1^0	0.38	0.57	1783888.1	0.42	0.77	1.61
s_1	20.87	24.96	26.03	1.67	1.10	32.68
t_1	220.91	258.00	18.80	18.79	1.44	713.68

[a] Systems as in Table 2.

[b] Fitted to the modified Eq. (3) discussed in the text.

this modification is used. A similarly interesting behavior of this system is observed when we apply the genetic algorithms. This issue is discussed below.

Apart from the global minimum, the Monte Carlo method described in [1] finds all possible local minima and pseudo-solutions. Thus, the retention surface of adenosine in acetonitrile–aqueous buffer mobile phase exhibits a local minimum with $\chi^2 =$ 4.40, which is found by the Monte Carlo method with a probability equal to 0.74, whereas the probability of finding pseudo-solutions is only 0.01 [1]. For the retention surface of adenosine in methanol–aqueous buffer mobile phase, the Monte Carlo method may be trapped in a local minimum with $\chi^2 = 10.19$ with a probability equal to 0.02 or in pseudo-solutions with the same probability of 0.02. The behavior of hydroxyzine and benzoic acid in methanol–aqueous buffer mobile phases is quite interesting; their retention surfaces do not exhibit local minima other than the global minimum. However, the Monte Carlo method may be trapped into a great number of pseudo-solutions with probability equal to 0.84 for hydroxyzine, if we do not use Eqs. (5), and 0.34 for benzoic acid. Finally, the retention surface of 5-hydroxyindole-3-acetic acid in isopropanol–aqueous buffer mobile phase exhibits a second minimum with $\chi^2 = 151.94$ found with a probability equal to 0.63 and a very small number of pseudo-solutions with a probability of 0.03.

Unlike the Monte Carlo method proposed in [1] that can be used straightforwardly, provided that we have defined proper ranges for the adjustable parameters, genetic algorithms require, apart from proper ranges, the optimization of a number of parameters, such as population size, maximum number of generations, type of mutation and mutation rate, type of crossover and crossover probability, variance edge value, etc. Here, for the ranges of the adjustable parameters we used the same ranges with those of the Monte Carlo technique, i.e. the ranges depicted in Table 3.

Preliminary extensive tests using all the genetic algorithms have shown that the type of crossover has a small effect on the obtained results and a value of the crossover probability of around 0.8 gives the best results. The type of mutation affects the results. In particular the Gaussian mutation should be preferred over the uniform one. The mutation rate also decisively affects the performance of a genetic algorithm. In order to find the optimum pair of crossover probability and mutation rate we worked as follows. The crossover probability was changed from 0.5 to 0.95 by steps of 0.025 and at each value of the crossover probability the mutation rate was changed from 0 to 0.5 by steps of 0.05, which were further reduced to 0.005 in the region of the best performance of the genetic algorithm.

The effect of the mutation rate on the performance of the classical algorithm applied to the first experimental system, i.e. to the retention surface of adenosine in acetonitrile–aqueous buffer mobile phase, is depicted in Fig. 1. Note that in Fig. 1 there are actually three different stacked figures, with three independent vertical axes. Each figure corresponds to a certain mutation rate. The *x*-axis is the run number of the genetic algorithm. That is, for example, the 20 points (○) shown in Fig. 1 are the solutions (χ^2 values) obtained by the classical genetic algorithm when it ran 20 times with mutation rate equal to 0.05. Note also that the retention surface of this data set exhibits two minima with $\chi^2 = 3.5$ (global minimum) and $\chi^2 = 4.4$. We observe that the optimum value of the mutation rate is 0.01. But even in this case the classical algorithm may be trapped into the local minimum $\chi^2 = 4.4$ with a probability of ca. 0.5.

The maximum number of generations used for the

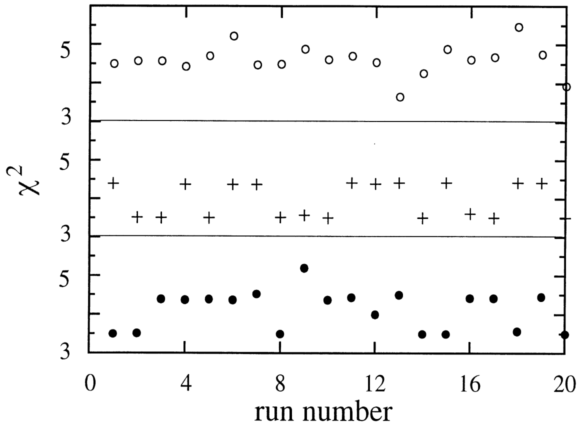

Fig. 1. Effect of mutation rate on the convergence of classical algorithm applied to adenosine in acetonitrile–aqueous buffer mobile phase using the following parameters: maximum number of generations = 10 000, population size = 100, crossover probability = 0.8 and Gaussian mutation with rate = 0.05 (○), 0.01 (+), 0.005 (●).

results of Fig. 1 was 10 000. We found that the obtained results are not improved if we increase this parameter. The results may be improved if we raise the population size. However, the increase of the population size is accompanied by a considerable increase of the computation time and for this reason we used populations of 100 chromosomes in all cases.

Since the classical genetic algorithm may be stuck in a local minimum other than the global one, we proceeded by examining the performance of Haupt's and Brunetti's algorithms using the same experimental system, that is the retention surface of adenosine in acetonitrile–aqueous buffer mixtures. The optimum values of the various parameters were the following: for Haupt's algorithm maximum number of generations = 10 000, mutation rate = 0.05 and $N_{\mathrm{good}} = 70$. For Brunetti's algorithm we used 30 000 generations, crossover probability = 0.8 and variance edge value = 0.01. Note that the performance of Brunetti's algorithm increases by increasing the maximum number of generations, and the value 30 000 was arbitrarily selected. The results obtained by these two genetic algorithms are similar to those depicted in Fig. 1 by points (+). That is, all the genetic algorithms we studied up to now can find the global minimum with a probability of around 0.45 but they may also get stuck to the local minimum $\chi^2 = 4.4$. Table 5 shows the probability P that an algorithm finds the global optimum, the mean value $\langle \chi^2 \rangle$, the standard deviation σ, and the minimum value of χ^2, χ^2(min), over a set of 20 values of χ^2 obtained by repeated runs of the genetic algorithm. In addition, the mean value of χ^2 and the standard deviation of the subset of the χ^2 values which are close to the global minimum ($\chi^2 < 4$) are also shown. We observe that the three genetic algorithms, i.e. classical, Haupt's and Brunetti's algorithms, exhibit almost the same performance.

It is seen that, contrary to Marques et al. [5] conclusion that a genetic algorithm is a global method, all the genetic algorithms we examined may be trapped into a local minimum other than the global one. For the retention surface of adenosine in acetonitrile–aqueous buffer mobile phase the genetic algorithms found the global minimum with a probability of ca. 0.45. This is better than that of the Monte Carlo method, but the time needed for the use of the genetic algorithms is much more than that of the Monte Carlo method. Thus, the Monte Carlo method requires only 4 min on a PC with a 600 MHz Pentium III processor. On the other hand, one run of the classical genetic algorithm requires 2 min and that of Brunetti's algorithm 6 min. Since we need about 20 runs for a safe picture of the global minimum, the computation time is 40 min for the classical algorithm and 2 h for that of Brunetti, plus the time needed for optimization of their parameters.

At this point we should clarify the following. A genetic algorithm does not converge to a certain value of χ^2, but to a spectrum of χ^2 values, which are close to the global or local minimum. Thus, the

Table 5
Results from the application of the genetic algorithms to systems 1 and 2[a]

Algorithm	P	$\langle \chi^2 \rangle$	σ	χ^2 (min)	$\langle \chi^2 \rangle$	σ
System 1						
	All values of χ^2				$\chi^2 < 4.0$	
Classical	0.50	3.96	0.45	3.5	3.52	0.04
Haupt	0.45	4.03	0.44	3.5	3.56	0.13
Brunetti	0.45	4.08	0.44	3.5	3.59	0.08
Our modification	1.00	3.55	0.05	3.5	3.55	0.05
System 2						
	All values of χ^2				$\chi^2 < 9.5$	
Classical	1.0	8.01	0.13	7.8	8.01	0.14
Haupt	0.6	11.94	8.04	7.9	8.05	0.13
Brunetti	0.8	9.18	2.83	7.9	8.20	0.41
Our modification	1.0	8.06	0.12	7.8	8.06	0.12

[a] Systems as in Table 2.

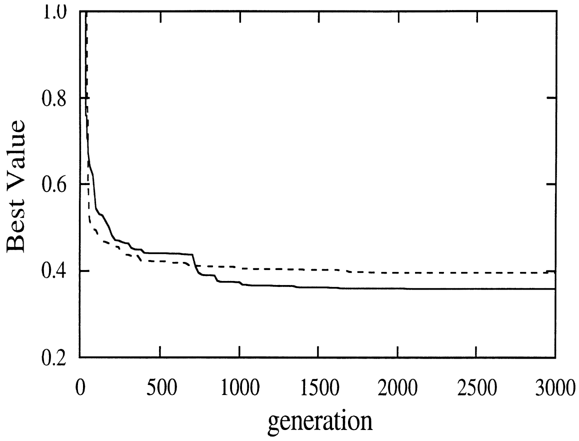

Fig. 2. Dependence of the best value upon generation for the classical algorithm of Fig. 1 in two different runs indicated by the solid and the broken lines using Gaussian mutation with rate 0.01.

classical algorithm converges to the global minimum $\chi^2=3.5$ with a probability ca. 0.5 means that only 50% of the χ^2 values obtained from the classical genetic algorithm are close to 3.5.

Working with the classical algorithm we found that beyond a certain maximum number of generations, say 2000 to 5000, there is no significant optimization in the convergence to the global or local minimum. This is clearly depicted in Fig. 2. Therefore, the system of adenosine in acetonitrile–aqueous buffer mixtures is ideal for applying the modification we propose.

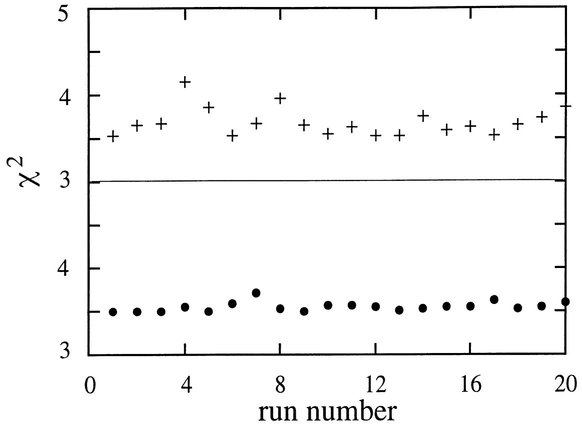

Fig. 3. χ^2 value vs. the run number for our modification of the classical genetic algorithm applied to adenosine in acetonitrile–aqueous buffer mobile phase using the following parameters: population size = 100, crossover probability = 0.8, Gaussian mutation with rate 0.01, and $m=5000$, $M=6$ (●); $m=3000$, $M=10$ (+).

We applied our modification using the same type and probability of crossover as well as type and rate of mutation with those of the classical algorithm. According to the modification we propose, the classical algorithm is run for m generations, then an entirely new population is generated with $N-1$ members and the best chromosome of the previous population is added as the N^{th} member. The whole procedure is repeated M times and the best chromosome of all these populations is the final solution. In our application we tested two alternatives: $m=5000$, $M=6$ and $m=3000$, $M=10$. The obtained results are depicted in Fig. 3 and in Table 5. It is seen that the choice $m=5000$ and $M=6$ gives the best results.

We observe that from the four genetic algorithms we examined only the modification we propose approaches to a global method. However, this feature is valid under the prerequisite that the classical genetic algorithm converges within a relatively small number of generations, less than 5000.

In order to examine whether this picture is characteristic of the particular system, i.e. of the retention surface of adenosine from acetonitrile–aqueous buffer mixtures, or not, we repeated the above analysis to the rest of the experimental systems adopted in the present study. The study of various systems, apart from the performance of the various algorithms, allows us to examine whether the parameters of the algorithms require adjustment in each system or not. This information is crucial if we take into account the time needed in order to succeed an acceptable performance of a genetic algorithm.

First we examined the retention surface of adenosine in methanol–aqueous buffer mobile phases. We found that if the adjustments are as in the previous system, the classical algorithm is trapped into the local minimum $\chi^2=10.19$ with a probability of 0.5, much higher than that of the Monte Carlo method, which is only 0.02. However, the results are improved radically by increasing the mutation rate to 0.02. In this case the algorithm finds only the global minimum ($\chi^2=7.824$) with some oscillation (Table 5).

For the same system Haupt's algorithm gave very poor results. Thus, we found it very difficult to optimize this algorithm. The best results shown in Table 5 were obtained using a mutation rate = 0.03 and $N_{\mathrm{good}} = 70$. In this case we obtained: $\langle\chi^2\rangle = 11.94$, $\sigma = 8.04$, $\chi^2(\mathrm{min}) = 7.91$, whereas the global minimum of this system is 7.824. The performance of Brunetti's algorithm is better than that of Haupt's algorithm but still poor (Table 5). The results of our modification with $m = 3000$ and $M = 10$ repetitions (total number of generations = 30 000) using mutation rate = 0.02 converge to the global minimum with 100% success exhibiting a small oscillation.

The results obtained for the last three systems, i.e. hydroxyzine and benzoic acid in methanol–aqueous buffer mobile phase and 5-hydroxyindole-3-acetic acid in isopropanol–aqueous eluent, are shown in Table 6. For the first two systems classical and Brunetti's algorithms were used with the same parameters as in the system of adenosine in methanol–aqueous buffer mobile phase. Our modification was applied with $m = 5000$ and $M = 6$. Haupt's algorithm was used with $N_{\mathrm{good}} = 70$ and a mutation rate equal to 0.06 for hydroxyzine and 0.07 for benzoic acid. It is seen that Haupt's algorithm gives the worst results, whereas all the other algorithms converge satisfactorily to the global minimum. However, the most interesting result from this table concerns the effect of the equation used in the fitting procedure on the obtained results. We observe that when we replace the fitted equation by a more appropriate one, in our case if we modify Eq. (3) using Eqs. (5) for the system of hydroxyzine in methanol–aqueous buffer mobile phase, then the performance of all genetic algorithms, including Haupt's algorithm, is increased considerably. For the system of 5-hydroxyindole-3-acetic acid in isopropanol–aqueous buffer mobile phase all the genetic algorithms give very good results except for Brunetti's algorithm whose convergence is rather poor. Note also that classical and Brunetti's algorithms were used with the same parameters as in the system of adenosine in methanol–aqueous buffer mobile phase, our modification was applied with $m = 3000$ and $M = 10$ and Haupt's algorithm was used with $N_{\mathrm{good}} = 70$ and mutation rate equal to 0.07.

We have seen that a genetic algorithm does not converge to the same solution. That is, different runs give different solutions, which may correspond to the same value of χ^2. This is clearly depicted in Table 7. However, this is not a problem since the distribution of values of a certain parameter falls within the uncertainty limits of this parameter. Thus if we compare the standard deviations of Table 7 to the corresponding values of Table 4 for system 1, we conclude that the uncertainty limits of each model parameter is about 10 times greater than those obtained from a genetic algorithm. For example, the values of k_1^0, which are obtained from the classical algorithm when $\chi^2 < 3.6$ for adenosine in acetonitrile–aqueous buffer mobile phase, have an average value equal to 0.95 and standard deviation 0.02. For this parameter from Tables 2 and 4 we have $k_1^0 = 0.925 \pm 0.38$.

It is seen that indeed the fact that a genetic algorithm yields a spectrum of solutions and model coefficients instead of a certain solution with certain model coefficients is not a serious weakness. However, if this feature of the genetic algorithms is considered as a defect, it can be completely overcome by combining a genetic algorithm with a nonlinear least-squares fitting procedure, like the Microsoft Solver, in the same way that the Monte Carlo

Table 6
Results from the application of the genetic algorithms to systems 3, 4 and 5[a]

Algorithm	System 3			System 3[b]			System 4			System 5		
	$\langle\chi^2\rangle$	σ	χ^2 (min)	$\langle\chi^2\rangle$	σ	χ^2 (min)	$\langle\chi^2\rangle$	σ	χ^2 (min)	$<\chi^2>$	σ	χ^2 (min)
Classical	1.306	0.310	1.068	1.070	0.0026	1.068	0.161	0.036	0.134	133.88	6.34	131.5
Haupt	4.482	2.636	1.068	1.070	0.0030	1.068	0.270	0.100	0.143	135.17	7.45	131.5
Brunetti	1.112	0.109	1.068	1.073	0.0050	1.068	0.163	0.026	0.127	145.58	19.69	131.7
Our modification	1.140	0.069	1.068	1.069	0.0006	1.068	0.134	0.007	0.127	133.07	4.62	131.4

[a] Systems as in Table 2.

[b] Fitted to the modification of Eq. (3) discussed in the text.

Table 7
Values of the adjustable parameters of Eq. (3), their average value, $\langle P \rangle$, and the standard deviations, σ, for adenosine in acetonitrile–water calculated from the classical genetic algorithm at five different runs

Solution	1	2	3	4	5	$\langle P \rangle$	σ
k_0^0	22.621	22.591	22.598	22.557	22.515	22.58	0.04
K_b^0	0.00033	0.000326	0.000327	0.000333	0.000333	0.00033	3.3×10^{-6}
q_1	23.670	27.663	27.029	24.294	20.769	24.69	2.78
s_0	−80.142	−80.107	−80.232	−80.115	−79.153	−79.95	0.45
q_2	−1375.92	−1437.12	−1428.54	−1392.69	−1328.78	−1392.61	43.66
t_0	712.092	712.371	715.242	714.856	689.699	708.85	10.80
k_1^0	0.945	0.991	0.957	0.927	0.951	0.95	0.02
s_1	−10.869	−15.807	−13.189	−10.255	−10.719	−12.17	2.33
t_1	50.394	102.407	77.010	47.291	47.610	64.94	24.36
χ^2	3.498	3.498	3.498	3.498	3.498	–	–

routine for search of initial estimates proposed in [1] is combined with the Solver. In this case the genetic algorithm searches for initial estimates and the nonlinear least-squares routine finds the final solution.

Here, we proceeded to this combination and in particular we combined the classical genetic algorithm and our modification with the Solver. In fact we used the results of the genetic algorithms as initial estimates for the Solver. This combination gave the following results: for systems that exhibit just the global minimum, like the systems of hydroxyzine and benzoic acid in methanol–aqueous buffer mobile phases, this technique finds the global minimum with 100% success. It is seen that it is superior to the Monte Carlo technique suggested in [1] if we ignore the parameter of time consumption. In systems with two or more minima the probability of finding the global minimum is equal to or slightly better than that of finding the global minimum by the genetic algorithm alone. Thus, the combination of our modification with the Solver finds the global minimum with a success close to 100%. At this point we should clarify again that the genetic algorithm we propose gives a spectrum of χ^2 values that in all cases we examined has an average value very close to the global minimum. The combination of this genetic algorithm with the Microsoft Solver finds the accurate value of the global minimum with a probability close to unity and from this point of view this combination should be preferred to the use of the genetic algorithm we propose alone.

We shall complete our discussion with some general comments. The investigation carried out in this paper was focused on the fitting problem itself using genetic algorithms. Solving the fitting problem is useful in three main respects: first, tables with chromatographic data can be replaced by a single equation; second, this equation may be used for optimizing separations on a certain column; and third, if this equation expresses a certain model, we may discuss about physical or chromatographic insights. Thus solving the fitting problem itself does not provide chromatographic insights. It is the equation to be fitted that does that. Eqs. (2) and (3) we adopted are widely used in chromatographic studies but they are not related to a strict model. They do not take into account activity coefficients [15,16] and they assume an empirical dependence of ln K upon φ, whereas the dependence of pK upon the modifier content in the mobile phase is quite complicated [17,18]. For this reason no discussion about chromatographic insights was made in the present paper. For the same reason we did not examine over-fitting problems. The over-fitting is a great problem irrespective of the purpose of the fitting procedure. When it appears, the fitting equation cannot be used for interpolation and/or its parameters have no physical content. However, the over-fitting is not related to the method used to fit an equation to a data set, as is evident from the fact that using genetic algorithms or the Solver we find the same global minimum. The over-fitting is due either to the use of an improper fitting equation or/and to the data set when there are very noisy data or data improperly spaced.

The time needed for the application of a genetic algorithm and the spectrum of solutions and model coefficients that it yields instead of a certain solution with defined model coefficients are usually considered as the most serious disadvantages of the genetic algorithms. Although the later drawback can be easily overcome by combining the genetic algorithm with a non-linear least-squares routine, as described above, we should have in mind that the main problem of fitting is not how much time we need to find a solution or if the solution will be a spectrum of solutions very close to the global minimum, but the certainty that the solution we found is indeed the global minimum. We shall make clear how difficult this issue is by an example. Consider that we are fitting the experimental data of system 1 to Eq. (2) using the Solver. In order to obtain good initial estimates the experimental data set is divided into four subsets, each subset characterized by a constant value of $\varphi = 0$, 0.01, 0.03 and 0.05. Then using the Solver each subset is fitted to the following equation:

$$k = \frac{k_0 K_b + k_1 10^{-\mathrm{pH}}}{K_b + 10^{-\mathrm{pH}}} \tag{6}$$

In this way the dependence of ln K_b, ln k_0 and ln k_1 upon φ is determined. Therefore, the initial estimates of the adjustable parameters of Eq. (2) are obtained by a least-squares fit of the ln K_b, ln k_0 and ln k_1 values to second order polynomials, since ln $K_b = \ln\ K_b^0 + q_1\varphi + q_2\varphi^2$, $\ln\ k_0 = \ln\ k_0^0 + s_0\varphi + t_0\varphi^2$ and $\ln k_1 = \ln k_1^0 + s_1\varphi + t_1\varphi^2$. Then we find: $K_b^0 = 0.00034$, $k_0^0 = 21.7$, $k_1^0 = 1.0$, $q_1 = -34.5$, $q_2 = 768$, $s_0 = -67.5$, $t_0 = 398$, $s_1 = 0.35$ and $t_1 = -455$. However, if these values are used as initial estimates, the Solver will converge to the local solution $\chi^2 = 4.5$ and not to the global one $\chi^2 = 3.5$. Note that the above procedure of finding initial estimates is the most rigorous and for this reason it gives the illusion that the solution $\chi^2 = 4.5$ corresponds to the global minimum of system 1. However, only by great chance we could find the global minimum of system 1 by using the Solver or any other non-linear least-squares routine without an additional routine for searching for initial estimates.

The above example clearly points out the need for more powerful methods for the fitting problem in cases where the fitting function has a large number of adjustable parameters. In [1] we have proposed a combination of a Monte Carlo technique and the Solver to solve the fitting problem. The above routine finds the global minimum of system 1 with a probability equal to 25% (Table 2). The improvement in finding the global minimum is considerable. However, extensive tests of the Monte Carlo method have shown that there exist cases that this method may fail to find the global minimum. For example, if we apply the Monte Carlo method to system 3 using the range 0 to 1 for the variable K_a^0, the method cannot find a solution. This is due to the fact that the Monte Carlo technique is based on a random search for initial estimates, which leaves a (small) probability of missing the global minimum in a ‘‘difficult’’ system, i.e. in a system where the global minimum is attained only when the initial estimates are very close to it. This drawback of the Monte Carlo search may be overcome by the use of a different search than the random one and since it is well known that the genetic algorithms use an ‘‘organized’’ search to find the global minimum, we investigated in the present paper the use of these algorithms to the fitting problem. The obtained results confirmed our choice. Thus, whereas the global minimum of system 1 is found with a probability about 25% using the Monte Carlo routine, this probability is close to 100% by using the genetic algorithm we propose. It is seen that indeed the proper use of the genetic algorithms increases considerably the confidence that the solution we find corresponds to the global minimum.

Finally, we should point out that in this paper we have adopted the least-squares criterion, Eq. (1), for the best solution. This is a criterion generally accepted when there are no outliers. Outliers are data points that are far away from the ‘‘true’’ values. Therefore, the residuals of the outliers are very large and since in the least-squares technique the residuals are squared, outliers have a significant contribution to χ^2. In the presence of outliers the least-squares technique should be replaced by a robust method. The genetic algorithms as well as the Solver can acquire high robustness if the least-squares criterion of Eq. (1) is replaced by the least sum of absolute deviations $|\chi|$ [19]. So the influence of the outliers is much less on $|\chi|$ than on χ^2. The experimental data we used here do not have outliers, as is easily detected from the plots of k vs. φ at constant pH and k vs. pH at constant φ. For this reason the least-squares technique was adopted here for finding the best solution. The high value of $\chi^2 = 131.4$ for system 5 is not due to outliers but to the fact that the values of k are much higher than those of the other systems. In addition, it might show that a better model than that of Eq. (3) should be more suitable for this system. However, the target of this investigation was not to find out which equation describes properly each experimental data set but to solve the fitting problem using genetic algorithms.

6. Conclusions

From the study carried out above the following conclusions can be drawn:

1. The four genetic algorithms tested in the present paper do not exhibit the same performance when they are used for the modeling of response surfaces in HPLC. The best results were obtained by our modification and the worst by Haupt's algorithm. The classical genetic algorithm gives satisfactory results better than those of Brunetti's algorithm.

2. The main disadvantage of the genetic algorithms is the rather great time needed for the optimization of their parameters. At this point Brunetti's algorithm has an advantage over the others, since it does not use mutations.

3. The above disadvantage lessens if we take into account that the optimum values of the various parameters of the genetic algorithms do not depend significantly on the particular system, provided that it belongs to the same class of systems. Thus, we found that for modeling retention surfaces in HPLC and for a population size equal to 100, the best crossover probability is 0.8, the type of crossover does not play an important role and the optimum mutation rate varies in the narrow range from 0.01 to 0.02 when Gaussian mutation is used.

4. The combination of a genetic algorithm with the Solver finds the global minimum with a probability much better than that of the Monte-Carlo technique suggested in [1]. This probability is close to unity if the system exhibits just one minimum. It is also close to unity in systems with two or more local minima provided that we use our modification with the Solver.

References

[1] P. Nikitas, A. Pappa-Louisi, Chromatographia 52 (2000) 477.
[2] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.
[3] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3rd edition, Springer, Berlin, 1996.
[4] R.L. Haupt, S.H. Haupt, Practical Genetic Algorithms, Wiley, New York, 1998.
[5] R.M.L. Marques, P.J. Schoenmakers, C.B. Lucasius, L. Buyden, Chromatographia 36 (1993) 83.
[6] P. Schoenmakers, R. Tijssen, J. Chromatogr. A 656 (1993) 577.
[7] A. Brunetti, Comput. Phys. Commun. 124 (2000) 204.
[8] A. Pappa-Louisi, X. Portokalidou, J. Liq. Chromatogr. Relat. Technol. 23 (2000) 505.
[9] P.J. Schoenmakers, S. van Molle, C.M.G. Hayes, L.G.M. Uunk, Anal. Chim. Acta 250 (1991) 1.
[10] P.J. Schoenmakers, N. Mackie, M.L. Marques, Chromatographia 35 (1993) 18.
[11] M.L. Marques, P.J. Schoenmakers, J. Chromatogr. 592 (1992) 157.
[12] G.D. Christian, W.C. Purdy, J. Electroanal. Chem. 3 (1962) 363.
[13] P. Schoenmakers, H.A.H. Billiet, L. de Galan, J. Chromatogr. 218 (1981) 261.
[14] K.A. Dill, J. Phys. Chem. 91 (1987) 1980.
[15] M. Roses, I. Canals, H. Allemann, K. Siigur, E. Bosch, Anal. Chem. 68 (1996) 4094.
[16] I. Canals, J. Portal, E. Bosch, M. Roses, Anal. Chem. 72 (2000) 1802.
[17] F. Rived, M. Roses, E. Bosch, Anal. Chim. Acta 374 (1998) 309.
[18] S. Espinosa, E. Bosch, M. Roses, Anal. Chem. 72 (2000) 5193.
[19] W.H. Press, B.P. Frannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in Pascal, Cambridge University Press, Cambridge, 1986.