Appl. Math. and Comp. Sci., 1998, Vol.8, No.4, 865–886

NEURAL NETWORK SIGNAL INTERPRETATION FOR OPTIMIZATION OF CHROMATOGRAPHIC PROTEIN PURIFICATIONS

ERIC J. KLEIN*, SHEYLA L. RIVERA**

Mobile phase pH and salt gradient steepness are optimized for the separation of protein mixtures using gradient elution ion-exchange chromatography. The optimization method utilizes a factorial experimental design to generate an experimental matrix. The resulting chromatographic peaks are classified into six distinct classes based on peak geometry by a vector quantizing neural network (VQN). A modified chromatographic optimization function (COF), which accounts for the neural net classification as well as peak separation and total analysis time, is used to rank chromatograms in order of desirability. Results of the COF analysis are fit to a second order polynomial model, which is optimized in the experimental parameters using an advanced simplex algorithm.

1. Introduction

High performance liquid chromatography (HPLC) is extensively used as one of the steps in the purification of biomolecules. HPLC is becoming an increasingly popular process industrially, especially in the pharmaceutical industry where separating relatively small concentrations of bioproducts from process solvents with repeatable accuracy is required. In the ion-exchange mode of HPLC, the concentration profile in the stationary phase, and thus the elution times and profiles of the solutes, is dependent on ionic strength and pH of the mobile phase. Elution is performed by introducing a linear gradient of increasing ionic strength to the column using a salt such as NaCl (Yamamoto *et al.*, 1983).

The selection of optimum operating conditions is difficult and complex for multicomponent protein mixtures. This is due to the lack of a robust process model as well as to the high degree of interaction between process variables. Optimum operating conditions would allow for an adequate separation of the feed slug in the shortest

* Stevens Institute of Technology, Department of Chemical Engineering, Hoboken, NJ 07030
Current address: University of Colorado, Department of Chemical Engineering, Campus Box 424, Boulder, CO 80309, e-mail: eklein@optimal.colorado.edu.

** Stevens Institute of Technology, Department of Chemical Engineering, Hoboken, NJ 07030
Current address: Frito-Lay, Inc. -Technology, 7701 Legacy Drive, Plano, TX 75024, e-mail: Sheyla.Rivera@Fritolay.com.

amount of time possible, thereby maximizing productivity and product purity while minimizing downstream processing costs.

Several process parameters must be investigated in this optimization problem. Some of these parameters such as mobile phase composition, flow rate, temperature and pH can be optimized on-line while others such as chromatographic mode, column packing and column size must be selected *a priori* based on the experience of the chromatographer. Our goal is to optimize process parameters that can be manipulated on-line, such as mobile phase composition and pH. It is assumed that the optimum mode of chromatography, as well as the optimum column for the separation, has been previously chosen from vendor product information or from previous chromatographic experiences.

In this study, the pH and slope of the salt gradient (ionic strength) are optimized simultaneously to achieve adequate separation of ternary and quaternary protein mixtures in an acceptable analysis time. This is a challenging optimization due to the complex interactions between these two process variables. Both of these variables have a strong effect on the elution times of the components, and therefore on the degree of separation between neighboring peaks and on the total analysis time. Changes in pH affect the proteins' affinity for the ionically charged column packing since the formal charge of the proteins is a function of pH. Changes in gradient slope also affect the elution times of the proteins due to the competitive binding between salt ions and proteins within the column.

HPLC methods development is traditionally accomplished through an exhaustive grid search experimental method. Though this method results in acceptable product separation, it is an extremely inefficient trial-and-error process. It is overly expensive due to both the time involved and the potential product wasted in the extensive experimentation.

A success has been reported in predicting chromatograms with theoretical models (Gallant *et al.*, 1995a; 1995b; Gu, 1995). These models are, however, generally system specific in that they include thermodynamic properties of the components in the feed mixture. While these models can be used to predict elution times of a known sample, they cannot be used in predicting chromatograms for unknown samples, which are frequently encountered. Furthermore, it is required that the kinetic properties of the system be measured or predicted in order to use these models. Experiments must be performed to estimate constants in the adsorption isotherm (usually the Langmuir isotherm) being used to model the nonlinear adsorption occurring in the column. These constants can be shown to be functions of the retention times of the eluents, which are known to be dependent on mobile phase composition and pH. Therefore these models cannot be used to predict chromatograms and optimize the separation without performing several experiments at each mobile phase composition investigated. This extensive experimentation causes the use of these models in process optimization to be prohibitive.

Several researchers have found that the use of a statistical factorial design test matrix and a criterion function coupled with an optimization strategy can reduce the number of experiments necessary in the optimization process. The optimization strategies employed previously include computerized grid searches (Lundell and

Markides, 1993; Cotton and Down, 1983), response surface mapping (Felinger and Guiochon, 1992; Palasota *et al.*, 1992; Wang *et al.*, 1991), and the sequential simplex algorithm (Berridge, 1985; Walters *et al.*, 1991).

The method presented in this paper consists of a factorial design, a neural network to provide automatic peak classification and the use of a response function to quantify experimental results. These response values are regressed to a quadratic model, which is subsequently optimized using a sequential simplex algorithm. This method proves to be extremely efficient, requiring only seven experiments to optimize the two factors investigated, namely pH and linear salt gradient steepness. Though this method is illustrated here for known protein mixtures, the methodology is based on the fact that no *a priori* knowledge of the mixture is necessary, allowing this optimization scheme to be valid for unknown as well as known feed mixtures.

2. Optimization Methodology

It is assumed that all fixed parameters are previously chosen based on the chromatographer's past experiences with similar sample mixtures. These fixed parameters include chromatographic mode, column size, packing type and size, etc. The Doehlert shell design, known for its high efficiency, is used to generate a test matrix which uniformly samples the experimental domain defined for the remaining variables such as temperature, flow rate, and mobile phase composition. The chromatographic peaks resulting from these experiments are classified based on peak geometry using a vector quantizing neural network (VQN).

A chromatographic optimization function (COF) is used to quantify each experimental separation based on VQN classification, peak separation, and total analysis time. COF values are fit to a quadratic response model using the singular value decomposition (SVD) algorithm. This response model is maximized in the experimental variables using a constrained, variable sized, sequential simplex algorithm. An additional experiment is performed at these "optimal" conditions, and the resulting COF is used to replace the worst COF in the matrix. Optimization can continue in this iterative manner until the chromatographer is satisfied with the results (Fig. 1).

2.1. Doehlert Shell Design

The Doehlert design has been previously used in the optimization of liquid chromatography (Hu and Massart, 1989; Bourguignon *et al.*, 1993). The Doehlert matrix is a hybrid factorial design which generates $d^2 + d + 1$ points which are equally spaced from each other in d-factor space (Doehlert, 1970). This factorial design, while still providing an adequate sampling of the entire experimental region, results in a significantly lower number of points than the conventional central composite factorial design which generates $2^d + 2d + 1$ points. This is especially true when several factors are involved in the optimization. The Doehlert matrix is the most efficient form of factorial design, with efficiency defined as the ratio of the number of experiments generated to the number of experimental variables.

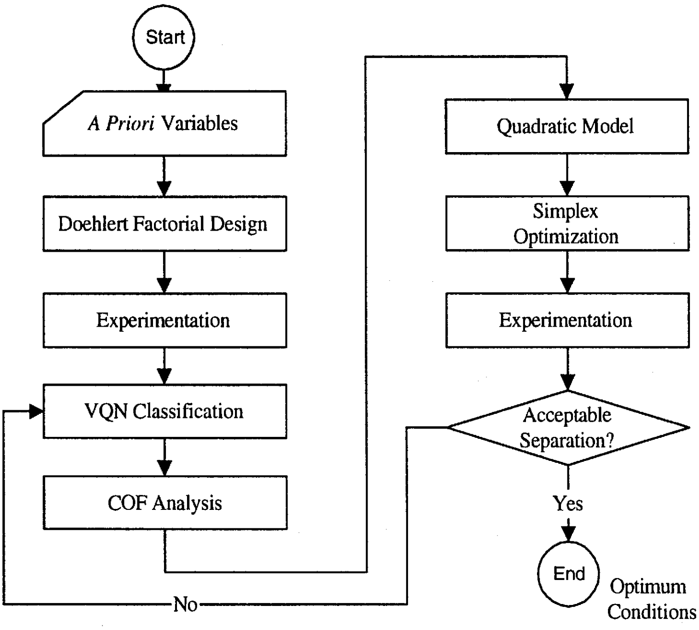

Fig. 1. Flow chart of optimization methodology.

Another attractive property of the Doehlert design is that it is multi-level. For example, in two-factor space, one factor (in this case pH) is subjected to five different values in the matrix while the other is subjected to only three (Fig. 2). This feature allows the researcher to subject the more significant process variables to more levels than other variables without altering the factorial design.

The Doehlert matrix is generated by first forming a simplex, one point of which is the center of the experimental domain. A simplex is a geometric figure having $d+1$ sides in d space. Therefore, for two factors, the simplex is an equilateral triangle. The remaining d^2 points are found by subtracting each point in the simplex from every other point.

In this study, the optimization variables are chosen to be the mobile phase pH and the slope of the linear salt gradient. Constraints are set on the optimization variables to define the experimental domain and the Doehlert matrix is generated (Table 1, Fig. 2). Low and high limits for mobile phase pH are set at 6 and 9 respectively, due to concerns of protein denaturation at extremely acidic or basic conditions. Limits of the salt gradient steepness are set at five and fifty column volumes (CV), where a column volume is a measure of time (i.e. volume of chromatographic column divided by mobile phase flow rate) commonly used in chromatography. The low constraint for gradient time is set at five CV since steeper gradients would approximate a step gradient rather than a linear gradient. The upper limit is set at fifty CV due to total analysis time considerations. The experiments are performed, and the resulting chromatographic peaks are classified using a vector quantizing neural network (VQN).

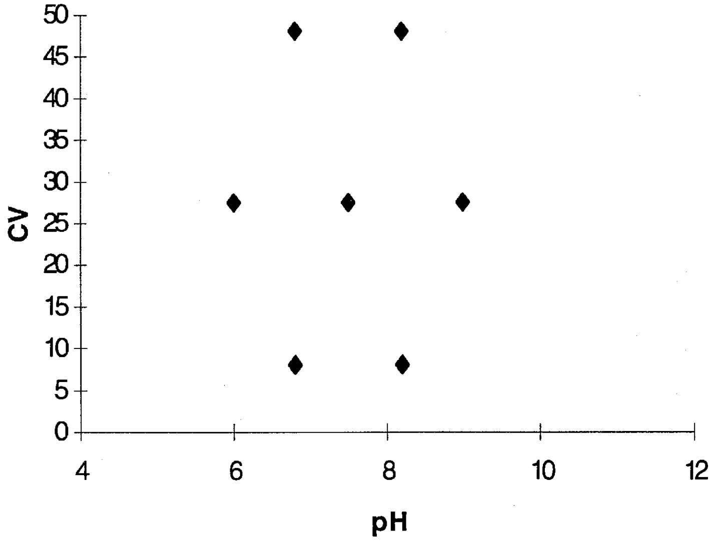

Fig. 2. Doehlert matrix.

Table 1. Results of Doehlert design.

Expt. #	1	2	3	4	5	6	7
pH	7.5	9.0	8.2	6.0	6.8	6.8	8.2
CV	27.5	27.5	47.0	27.5	8.0	47.0	8.0

2.2. Automatic Peak Classification

The task of automatic peak classification is accomplished by using a neural network methodology. Specifically, a vector quantizing network (VQN) is employed to classify peaks as either resolved or unresolved. A neural network is chosen for the task of pattern recognition since neural nets can be taught to recognize correlative patterns between input data and target values (Bhagat, 1990). VQN's have recently been extensively used as pattern recognition tools. Though VQN's have not been used previously to classify chromatographic peaks, they have been used in the fields of speech recognition (Huo and Chan, 1995), process control (Hinde and Cooper, 1993; 1994; Cooper *et al.*, 1992), and data compression (Gray, 1984).

The vector quantizing network (VQN) is employed to classify the chromatographic peaks into six distinct classes. These classes are Gaussian peaks, fronted peaks, tailed peaks, peaks with a leading shoulder, peaks with a trailing shoulder, and overlapping peaks (Fig. 3). Gaussian-shaped peaks are preferred while overlapping peaks result from an unacceptable chromatographic separation. Tailed and fronted peaks suggest a high concentration of solvent, and may require further downstream processing to reach purity specifications. Shouldered peaks may indicate co-elution.

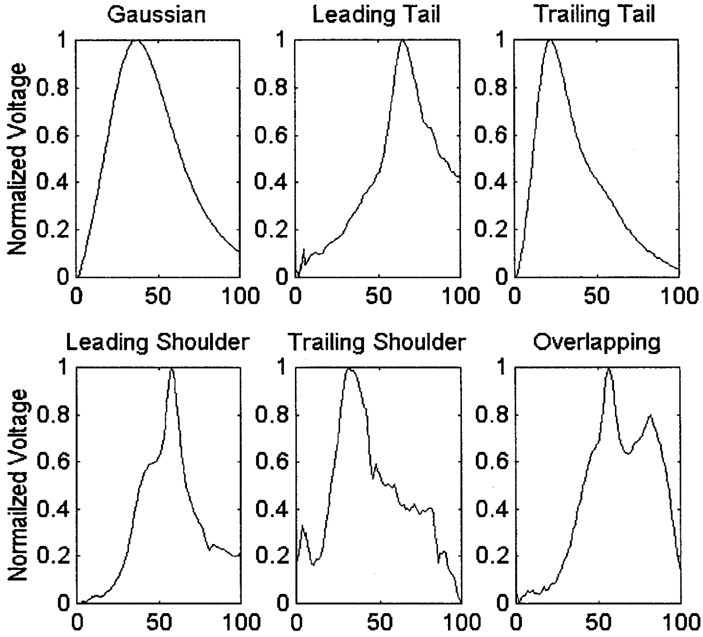

Fig. 3. Six peak classes recognized by the VQN.

The input to the VQN is a normalized vector representation of the chromatographic peak to be classified. The output of the VQN is a six element vector, containing all zeros except for one element which contains a value of unity. The element containing the value of unity corresponds to the class of the input peak as determined by the network.

2.2.1. Preprocessing

As a rule, size and shape of chromatographic peaks vary since peak areas are proportional to the concentrations of their respective components in the feed mixture. However, it is necessary that all digitized peaks (vectors) fed to the vector quantizing network be of identical size.

A preprocessing algorithm has been developed to map peaks to vectors of identical sizes so that they may be classified by the VQN. This preprocessing algorithm also filters out measurement noise and normalizes peaks so that peaks of similar geometry but different magnitude can be correctly classified (Fig. 4). Data acquisition

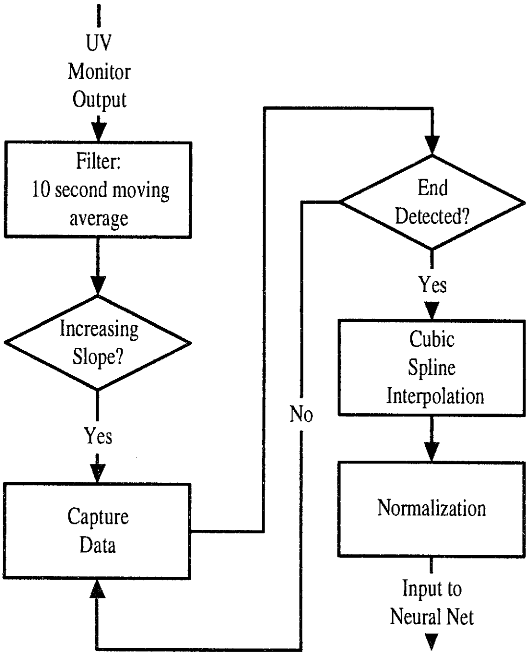

Fig. 4. Flow chart of preprocessing algorithm.

is facilitated by interfacing the output of the UV detector of the chromatographic apparatus with a personal computer. A ten second moving average is employed to filter out inherent measurement noise.

The presence of peaks in the UV signal is determined based on the percent change in the signal slope. A threshold in slope increase is set such that a continued slope increase above the threshold is recognized as the beginning of a peak. The ends of peaks are determined in a similar manner. In the case of overlapping peaks, indicated by the UV signal not returning to baseline between successive peaks, the overlapping peaks are captured as a single vector so that they may be identified as overlapping peaks by the VQN.

In the case of three or more successively overlapping peaks, neighboring peaks are captured in one vector, two peaks at a time, in a moving window fashion. The start of the first peak in the vector and the end of the second peak are artificially continued to the baseline by using the average values of the peaks' respective slopes. Thus the final result is that vectors representing overlapping peaks fed to the network contain two peaks, with the first peak starting and the second peak ending at baseline level. Data used to train the network is preprocessed in a similar manner. This is necessary since the extraordinary amount of data which would be needed to train the VQN to recognize all convolutions of higher numbers of overlapping peaks is prohibitive.

Upon identification of the beginning of a peak, the UV monitor output voltage is saved as a function of time in a matrix at the rate of one measurement per second. Voltage is recorded until the end of the peak is identified. The two columns of this resulting matrix generally contain between one hundred and five hundred elements.

This long voltage vector is then normalized with respect to time. Cubic spline interpolation is used to map the output voltage vector to a new vector containing exactly one hundred elements. Briefly, in cubic spline interpolation a third-order polynomial is used to interpolate between successive data points, with a unique polynomial being found between each successive pair of data points (Linfield and Penny, 1995).

The resulting voltage vector is then normalized with respect to its own maximum voltage so that peaks of varying magnitudes can be compared. This normalized, digitized peak, represented by one hundred data points ranging in magnitude from zero to one, is now ready to be fed to the vector quantizing network for analysis. Figure 5 contains plots of a single peak used to train the VQN in three different stages of this preprocessing algorithm. As can be seen, no peak geometry information is lost as a result of preprocessing.

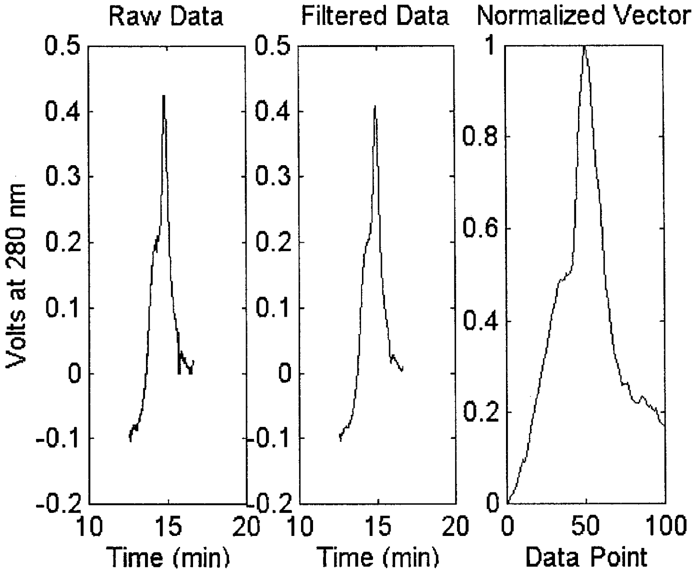

Fig. 5. Example peak at several stages of preprocessing algorithm.

2.2.2. Vector Quantizing Network

Vector quantizing networks (VQN) have been previously used for pattern recognition and classification, though not for the classification of chromatographic peaks. VQN's consist of three layers of neurons: an input layer, a competitive layer, and a linear layer (Fig. 6) (Linde *et al.*, 1980; Demuth and Beale, 1994). The input to the VQN contains R nodes, where R is the number of elements in each input vector. In this study, R was set to one hundred.

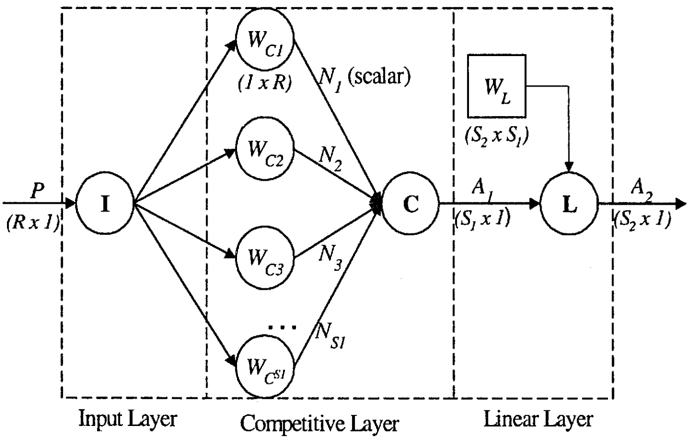

Fig. 6. VQN block diagram.

The input layer, **I**, distributes the input vector to each neuron of the competitive layer. That is, each vector fed into the input layer is output to each neuron of the competitive layer allowing each competitive neuron to receive the entire input vector **P**, which is a column vector of size R.

The competitive layer is the site of the actual pattern recognition. During network training, a unique weight vector is stored in each competitive neuron. These vectors represent the patterns each neuron is trained to recognize and comprise the weight matrix, $\mathbf{W_C}$, which is of size $S_1 \times R$ where S_1 is the number of competitive neurons. In this study, S_1 is set to eighteen, with three representative peaks from each of the six classes being used to train the VQN.

The first calculation performed by the competitive layer is a distance calculation. That is, the distance between each competitive neuron's stored weight vector and the input vector **P** is calculated:

$$N_i = -\left\|\mathbf{W_{c,i}} - \mathbf{P}\right\|_2 \tag{1}$$

where $i = 1, \ldots, S_1$, N_i is the negative of the distance and $\mathbf{W_{c,i}}$ is the weight vector of competitive neuron i. The outputs N_i are scalar quantities and range from slightly negative numbers to zero, with a value of zero indicating a perfect match between the input vector **P** and competitive neuron i's weight vector.

The second calculation performed by the competitive layer in node **C** is a sorting calculation to find the maximum value of N_i. The output vector $\mathbf{A_1}$, of length S_1, then contains a value of unity in one position, corresponding to the neuron with the maximum value of N_i. This is the neuron whose stored weight vector best matches the input vector. The remaining elements of $\mathbf{A_1}$ contain zeros.

The linear layer of the VQN, node **L**, combines the several competitive neurons into a few user-specified classes. The weight matrix $\mathbf{W_L}$ is of size $S_2 \times S_1$, where S_2 is the number of target classes. It is comprised of zeros and ones, with a one in matrix element $W_{L_{ij}}$ indicating that competitive neuron j is a member of class i. The linear layer performs the following calculation:

$$\mathbf{A_2} = \mathbf{W_L} \times \mathbf{A_1} \tag{2}$$

where $\mathbf{A_2}$ is the linear layer output vector of size S_2.

The result of this calculation is a value of unity in the element of $\mathbf{A_2}$ which corresponds to the class of the competitive neuron whose weight vector best matched the input vector. The remaining elements of $\mathbf{A_2}$ contain zeros.

2.2.3. Network Training

The vector quantizing network (VQN) is trained with eighteen experimental chromatographic peaks, with each of the six classes being represented by three peaks. These peaks are filtered and normalized with the aforementioned preprocessing algorithm prior to training the network. Examples of the peaks used to represent each class can be found in Fig. 3 while a summary of the target classes can be found in Table 2.

Table 2. Description of VQN target classes.

Class #	Description
1	Gaussian
2	Leading Tail
3	Trailing Tail
4	Leading Shoulder
5	Trailing Shoulder
6	Overlapping

The training of the VQN is supervised, with the weight matrix $\mathbf{W_L}$ being specified by the operator to assign each of the competitive neurons to the desired class. As discussed above, the matrix $\mathbf{W_L}$ is comprised of zeros and ones, with a one in the element $W_{L_{ij}}$ indicating that competitive neuron j is a member of class i.

The remainder of the training consists of training the competitive weight matrix $\mathbf{W_C}$. This is accomplished by feeding training data and corresponding target values to the network in a random order, and updating the column of $\mathbf{W_C}$ which corresponds to the competitive neuron with the highest output value accordingly. Thus, only the weights for the neuron best matching the given training vector are updated with each successive iteration.

Matrix $\mathbf{W_C}$ is initialized by setting the elements of each column to the midpoint of the highest and lowest elements found in the corresponding element of the vectors in the training set. Training then proceeds as follows: one by one the training vectors and their corresponding target classes are fed to the network in random order, and the neuron whose weights closest match the input vector is updated. If that neuron's target value (corresponding weight in matrix $\mathbf{W_L}$) is one, the neuron's weights move closer to the input pattern:

$$W_{C_{i,j}}^{k+1} = W_{C_{i,j}}^{k} + lr \times A_{1_i} \times \left(P_j - W_{C_{i,j}}^{k}\right) \tag{3}$$

where $i = 1, \ldots, S_1$, $j = 1, \ldots, R$ and lr is the learning rate, which is set to 0.05 in this study.

If the neuron's target value is zero, the neuron's weights move away from the input pattern:

$$W_{C_{i,j}}^{k+1} = W_{C_{i,j}}^{k} - lr \times A_{1_i} \times \left(P_j - W_{C_{i,j}}^{k}\right) \tag{4}$$

Bias terms are added to the outputs of the competitive layer, N_i, to ensure that all neurons are fired equally during training so that "dead" neurons, or neurons which are never updated during the training procedure, are avoided. These bias terms are updated after each iteration, increasing in value for neurons which were not fired and decreasing in value for neurons which were updated during that particular iteration (Klein, 1997).

In this study, three thousand iterations were required to provide good classification. While this may seem excessive, this requires less than five minutes of computing time on a standard Pentium personal computer and only needs to be performed once.

2.3. Chromatographic Optimization Function

In order to optimize the chromatographic separations, it is necessary to assign a numerical value to each separation performed so that the experiments may be ranked in order from most to least desirable. Several chromatographic response functions (CRFs) have been used previously. An extensive listing of these response functions was compiled by Berridge (Berridge, 1985). The most widely used of these functions is the resolution function:

$$R_{ij} = \frac{2(t_j - t_i)}{w_i + w_j} \tag{5}$$

where R_{ij} is the resolution between peaks i and j, t_i is the elution time of peak i, and w_i is the baseline band width of peak i (Fig. 7).

The individual peak resolutions can be summed to give an overall resolution for the chromatogram. However, the response function of eqn. (5) does not account for total analysis time, nor does it include a penalty for cases when more or fewer peaks than expected are eluted. This information is extremely important when dealing

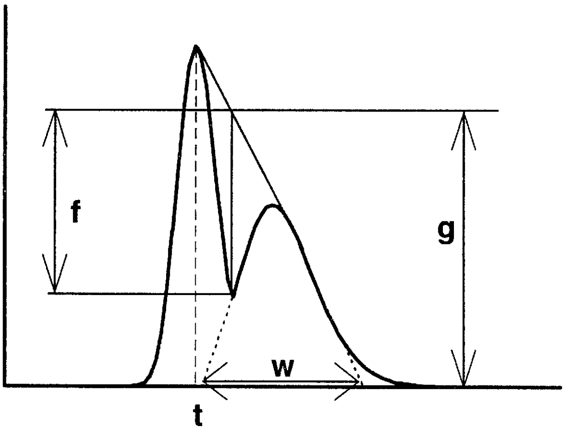

Fig. 7. Definition of COF parameters.

with an unknown sample, and it is desirable to incorporate this data in the response function. That is, there must be a way to penalize responses which do not contain as many peaks as other responses. This assures that all peaks are fully resolved and that peaks did not co-elute, especially if the number of expected peaks is unknown.

The response function chosen for use in this study is a modification of the combination of two functions reported by Berridge (1985). The chromatographic optimization function (COF) used has the form:

$$\mathrm{COF} = \sum_{i=1}^{np} \left[\ln\left(f_i/g_i\right)\right] - A(M-N) + B(t_m - t) + \sum_{i=1}^{n} K_i \tag{6}$$

where f_i and g_i are parameters used to describe the resolution of peak pair i (Fig. 7), np is the number of peak pairs, M is the number of expected peaks, N is the number of peaks eluted, t_m is the maximum desirable total analysis time, t is the elution time of the last peak, K_i is the penalty based on neural network peak classification, n is the number of peaks, and A and B are user-adjustable weights.

The first term of the COF describes the actual separation of the chromatographic peaks. It is desirable that f be equal to g (i.e. baseline separation). Therefore, the first term is a negative number which approaches zero as the separation becomes ideal. The second term accounts for cases in which some chromatograms contain more or fewer peaks than others. This term subtracts a penalty from the COF value for chromatograms which contain fewer than the expected number of peaks (i.e. in chromatograms where peaks co-elute). In unknown separations, M can be set to the maximum number of peaks found in the experiments, or it can be set to an arbitrary value thereby penalizing all chromatograms equally. The third term accounts for total analysis time, which is desirable to minimize in the optimization. The maximum desired analysis time, t_m, is chosen to be 15 CV. The adjustable weights, A and B, are set to 2 and 0.1, respectively, so that each term in the chromatogram will be of the same order of magnitude, namely that of unity.

The final term in the COF assesses a penalty based on the neural network classification of the peaks (Table 3). Classes 2 and 3, tailed peaks, are assigned a penalty of -0.1 while classes 3 and 4, peaks with shoulders, are assigned a penalty of -0.25. Tailed peaks are undesirable since they contain a high concentration of solvent and may require further processing. Shouldered peaks are less desirable than tailed peaks since a shoulder could indicate a hidden, or unresolved, peak. Classes 1 and 6 are not penalized since class 1 is desirable and overlapping peaks (class 6) are already accounted for in the first term of the COF. The COF is maximized during the optimization process.

The magnitude of the penalties assessed based on network classification are determined as follows. It is desirable that tailed peaks be penalized less than shouldered peaks, yet extensively overlapping peaks should still be assessed the highest penalty by the first term in the COF (eqn. (6)). It is decided that tailed peaks are as undesirable as slightly overlapping peaks with an f/g value of approximately 0.9 while shouldered peaks are comparable to overlapping peaks with f/g values of approximately 0.75. The penalties of -0.1 for tailed peaks and -0.25 for shouldered peaks are found by taking the natural logarithms of these f/g values, analogous to the first term in eqn. (6).

Table 3. K_i values based on VQN classification.

Class	Description	K_i
1	Gaussian	0.00
2	Leading Tail	−0.10
3	Trailing Tail	−0.10
4	Leading Shoulder	−0.25
5	Trailing Shoulder	−0.25
6	Overlapping	0.00

2.4. Response Model

The resulting values of the chromatographic optimization function (COF) are fit to a quadratic model which is dependent on both pH and gradient column volumes (CV). The model is of the form:

$$y = a_1 + a_2x_1 + a_3x_2 + a_4x_1^2 + a_5x_2^2 + a_6x_1x_2 \tag{7}$$

where y = COF, x_1 = pH, and x_2 = CV.

Models of this type have been frequently reported in the literature as being useful for modeling chromatographic responses. It has been found that higher-order variable interactions are small (Lindberg *et al.*, 1981), and are therefore omitted here. Omitting higher-order interactions also significantly decreases the number of coefficients which must be calculated. In this study, there are six coefficients and seven experiments. Thus, enough data points still exist to perform a least-squares regression. Though more data points would provide for a more accurate fit of the model, the goal in this study is to optimize the system with a minimal amount of experimentation.

2.5. Simplex Optimization

The constrained, variable sized, sequential simplex algorithm known as the Nelder and Mead algorithm is employed to numerically optimize the above model within the experimental domain. A simplex is comprised of $d+1$ points in d factor space. An initial simplex is formed, and the objective function (in this case eqn. (7)) is evaluated at each of these $d+1$ experimental points. The responses are ranked in order of desirability. The lowest-ranking point is discarded from the optimization and a new point is calculated to replace it. This new point is obtained through a series of logical calculations with the simplex being able to reflect (R), contract (C), or expand (E) from the original point, and change direction accordingly (Fig. 8). In this way, the simplex "climbs" through the factor space seeking higher (or lower) values of the objective function until an optimum is reached (Fig. 9) (Walters *et al.*, 1991).

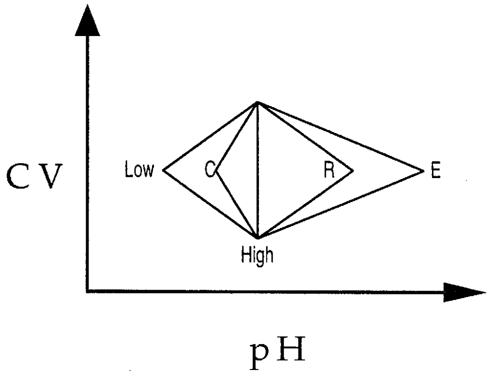

Fig. 8. Possible simplex movements.

It should be noted, however, that this algorithm very often returns local rather than global maxima. This problem is alleviated by starting the simplex algorithm at several different points which represent a broad range of the experimental domain. If all of these searches end at the same maximum, it is likely that this is a global rather than a local maximum for the region of interest. In this study, the simplex algorithm is started with five different starting simplexes to adequately explore the entire experimental domain.

The simplex method has been previously used in chromatographic optimizations. However, in some of these cases the values of the vertices have been evaluated through direct experimentation (Felinger and Guiochon, 1992; Palasota *et al.*, 1992). This is not recommended, as this can lead to excessive experimentation. Wang has proven

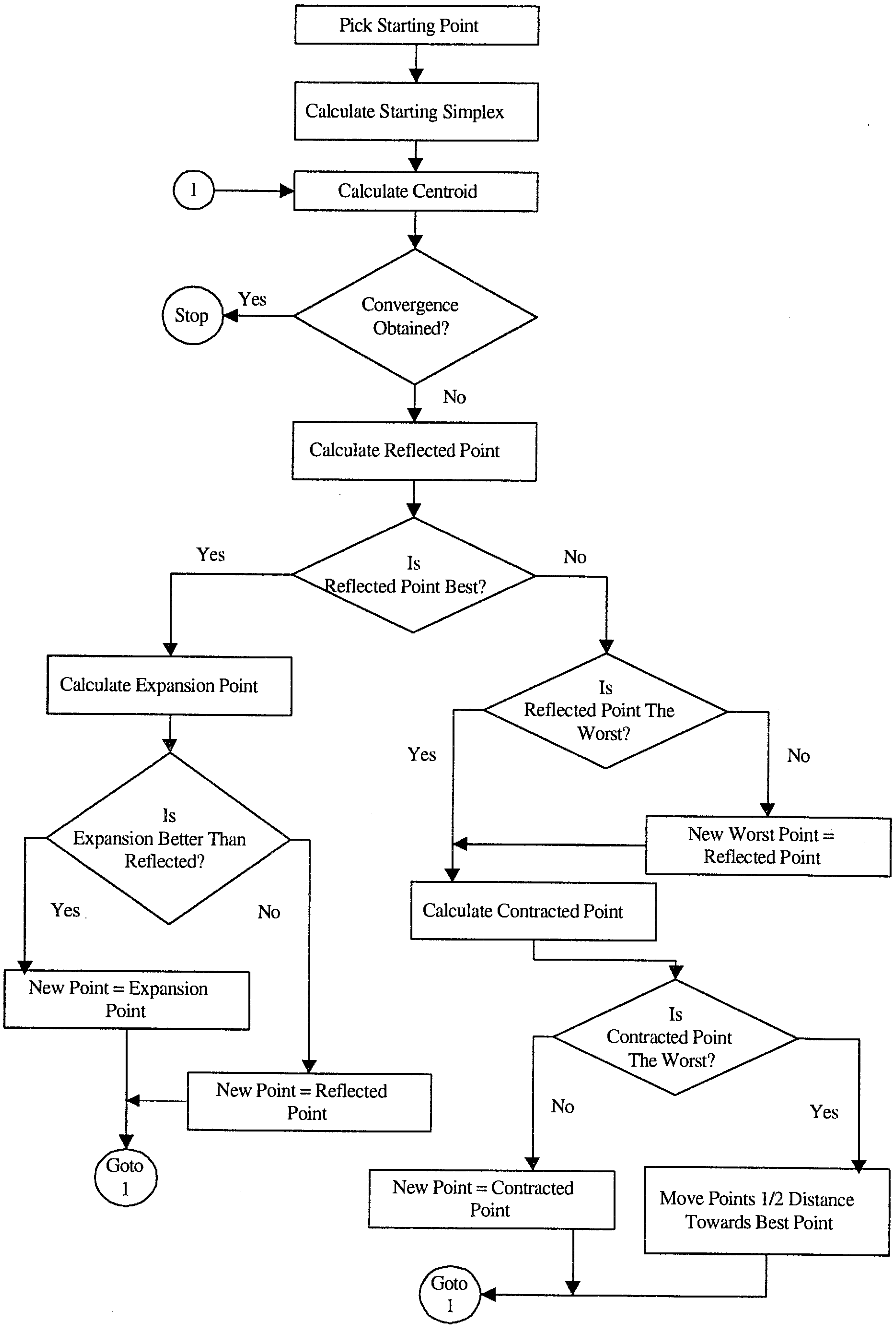
Pick Starting Point
Calculate Starting Simplex
1
Calculate Centroid
Stop
Yes
Convergence Obtained?
No
Calculate Reflected Point
Yes
Is Reflected Point Best?
No
Calculate Expansion Point
Is Reflected Point The Worst?
Yes
No
Is Expansion Better Than Reflected?
New Worst Point = Reflected Point
Yes
No
Calculate Contracted Point
New Point = Expansion Point
Is Contracted Point The Worst?
New Point = Reflected Point
No
Yes
Goto 1
New Point = Contracted Point
Move Points 1/2 Distance Towards Best Point
Goto 1

Fig. 9. Nelder and Mead simplex algorithm.

that it is more efficient to perform a small number of experiments and optimize a response model of the system as we have done here (Wang *et al.*, 1986; 1991).

3. Experimental Method

Equipment: The chromatographic setup consists of two Waters 510 HPLC pumps (Milford, MA), a Waters 590 programmable HPLC pump, a Waters automatic gradient controller, a Waters 440 UV detector operating at 280 nm, and a Waters 746 data module. The signal from the UV detector is also logged to a data file on a Gateway 2000 486 PC using PC-Lab Card's PCL-812PG data acquisition card and VisSim software (Visual Solutions; Westford, MA). The column is a weak anion exchanger PEEK fractogel column with DMAE chemistry (EM Separations; Gibbstown, NJ).

Mobile Phase Preparation: The mobile phase consists of 2 buffers (Sigma, LTD; St. Louis, MO), which are mixed to the desired mobile phase composition by controlling the flow rates of their respective pumps. Buffer A contains 10 mM Tris-[hydroxymethyl]amino-methane (TRIS) and 10 mM 1,3-bis[tris (hydroxy-methyl)-methyl-amino]propane (BIS). Buffer B contains 10mM TRIS, 10mM BIS, and 1M NaCl. Buffer pH is adjusted off-line to the desired pH of each run using 1M HCl and NaOH. The initial and final concentrations of the salt gradients in all experiments are 0.0M and 0.5M, respectively. All buffers are dissolved in deionized water, filtered through a 40 μm Millipore filter, and degassed with helium prior to use. The mobile phase flow rate is 1 ml/min in all experiments.

Sample Preparation: The proteins included in the sample are bovine serum albumin (BSA), lysozyme (L), conalbumin (C), and α-chymotrypsinogen A (A) (Sigma; St. Louis, MO). All proteins are dissolved in deionized water to concentrations of 6 mg/ml. Equal volumes of each standard protein solution are combined to give the final sample mixture. The sample size in all experiments is 25 μL.

4. Results

4.1. Case Study # 1

The protein mixture investigated in the first case study was a ternary mixture containing equal amounts of lysozyme (L), conalbumin (C) and bovine serum albumin (BSA). The Doehlert matrix experiments were performed and the resulting chromatographic peaks were classified by the vector quantizing network (VQN). A few representative examples of the peaks which were classified can be found in Fig. 10. Values of the chromatographic optimization function (COF, eqn. (6)), were determined for each experiment (Table 4). It should be mentioned that Experiment #1 of the Doehlert matrix (the central point in Fig. 2) was repeated after Experiment #7 to check for system degradation. The results of this second experiment were identical to those of Experiment #1.

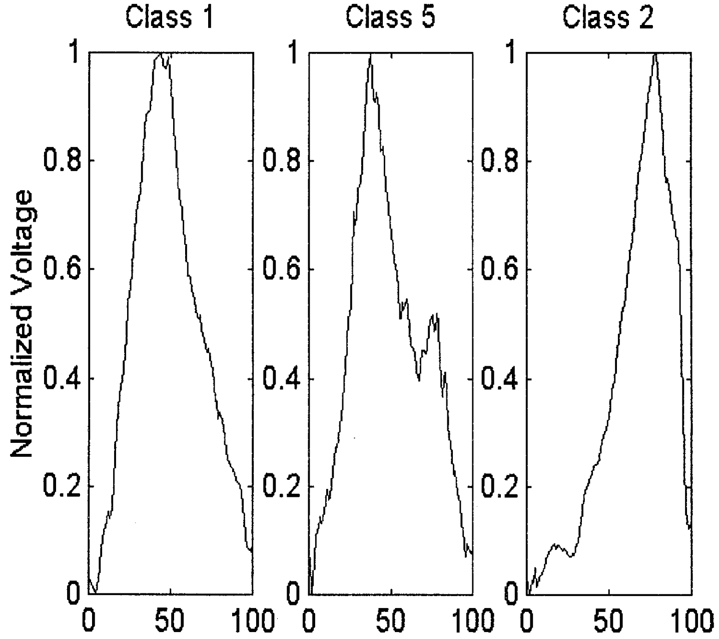

Fig. 10. Results of VQN classification.

The results of the data analysis were regressed to the quadratic model, eqn. (7). Statistical results of the regression are reported in Table 4 and model coefficients are reported in Table 5. In Table 4, SSE is the sum of the squared errors. The goodness-of-fit was evaluated using the incomplete gamma function, and is a measure of whether or not discrepancies between model results and experimental results are the result of chance fluctuations. Goodness-of-fit values greater than 0.1 suggest a believable model (Press *et al.*, 1986). Only 7 data points were used to determine the 6 coefficients of the model. Though more experiments will provide for a better fit of the data, this will also lead to a longer optimization time.

The quadratic model was optimized with the simplex algorithm. The optimum pH was found to be 8.8 and the optimum gradient steepness was found to be 48.6 column volumes (CV). These values correspond well with a visual inspection of the response surface of the model (Fig. 11).

A final experiment was performed at the optimum conditions. This confirmed that both baseline separation of all solutes and a short total analysis time occur at the conditions found by the simplex algorithm to be the optimum (Fig. 12).

4.2. Case Study # 2

The protein mixture investigated in the second case study was a quaternary mixture containing equal amounts of lysozyme (L), conalbumin (C), bovine serum albumin

Table 4. Case study #1: Experimental vs. model results.

Expt. #	pH	CV	COF	COF	Error (%)
			Actual	Model	
1	7.5	27.50	−4.57	−6.09	33.18
2	9.0	27.50	−1.02	−0.76	25.05
3	8.2	47.00	−3.83	−3.25	15.15
4	6.0	27.50	−12.46	−11.25	9.68
5	6.8	8.00	−2.04	−2.13	4.44
6	6.8	47.00	−10.54	−11.10	5.35
7	8.2	8.00	−0.38	−0.19	48.71
mean error = 20.22%		SSE = 4.52		goodness-of-fit = 0.48	

Table 5. Case study #1: Model parameters.

Coefficient	Value
a_1	−5.79E−03
a_2	1.57E−03
a_3	−1.24E+00
a_4	3.43E−02
a_5	5.00E−03
a_6	1.08E−01

(BSA) and α–chymotrypsinogen A (A). This separation was significantly more challenging than the first case study due to the extra protein in the mixture and the similarity among the retention times of C, A, and BSA. The optimization algorithm successfully optimized this difficult separation.

Statistical results and model parameters are again reported (Tables 6 and 7). The optimum experimental conditions were found to exist at pH 9.0 and 22.2 CV. Figure 13 contains the chromatogram produced at these optimum conditions. It should be noted that the rise in baseline at the end of the experiment was caused by the increasing salt concentration of the mobile phase.

5. Conclusions

A novel approach to optimal HPLC methods development is proposed for the ion-exchange separation of protein mixtures. The pH and ionic strength of the mobile phase are simultaneously optimized to maximize protein separation while decreasing total analysis time.

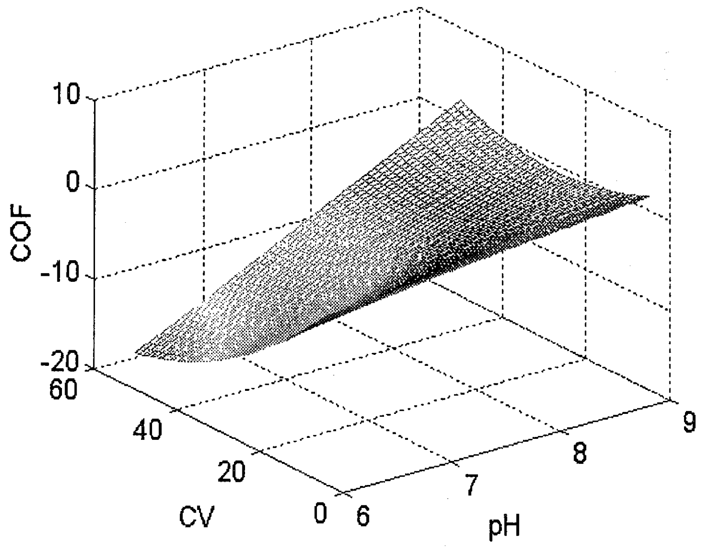

Fig. 11. COF response surface for case study #1.

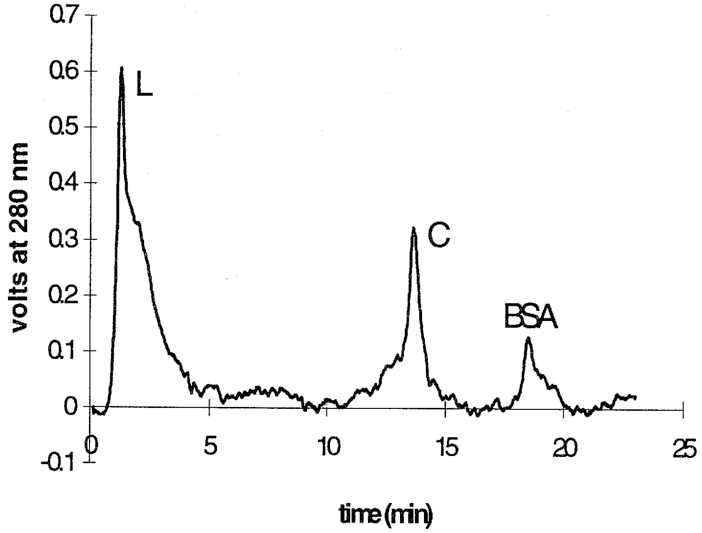

Fig. 12. Chromatogram at maximum COF, case study #1, pH = 8.8, CV = 48.6.

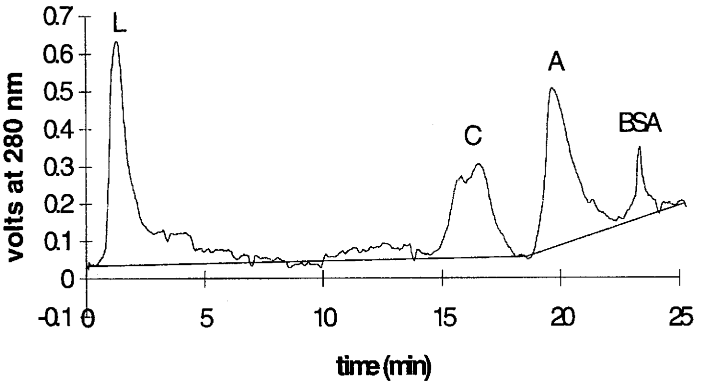

Fig. 13. Chromatogram at maximum COF, case study #2, pH = 9.0, CV = 22.2.

Table 6. Case study #2: Experimental vs. model results.

Expt. #	pH	CV	COF	COF	Error (%)
			Actual	Model	
1	7.5	27.50	−2.83	−3.47	22.42
2	9.0	27.50	−0.65	−0.76	17.76
3	8.2	47.00	−2.03	−1.55	23.68
4	6.0	27.50	−6.37	−5.64	11.40
5	6.8	8.00	−2.46	−2.73	11.19
6	6.8	47.00	−4.18	−4.66	11.32
7	8.2	8.00	−1.61	−1.29	19.60
mean error = 16.76%		SSE = 1.59		goodness-of-fit = 0.90	

Table 7. Case study #2: Model parameters.

Coefficient	Value
a_1	−2.56E−01
a_2	−9.86E−01
a_3	−3.80E−01
a_4	1.18E−01
a_5	2.24E−03
a_6	3.05E−02

A neural network peak classification procedure is developed and used to automatically categorize chromatograms. Specifically, a vector quantizing network (VQN) is used in conjunction with a preprocessing algorithm to classify chromatographic peaks into six distinct classes based on geometry. The diagnostic network algorithm is combined with the optimization methodology by accounting for peak classification in the chromatographic optimization function (COF). The COF is found to be useful in representing the quality of the protein separations.

A Nelder and Mead simplex algorithm is utilized in conjunction with a quadratic response model to determine the optimal operating conditions. Results are promising, and the optimization methodology will be tested on more challenging separations in the future. It should be emphasized that since this optimization scheme requires no *a priori* knowledge of the components in the feed, it can be applied to both known and unknown feed mixtures.

Acknowledgment

This work was partly funded by the National Science Foundation Research Grant No. BES–9411138.

References

Berridge J.C. (1985): *Techniques for the Automated Optimization of HPLC Separations.* — New York: John Wiley and Sons.

Bhagat P. (1990): *An introduction to neural nets.* — Chem. Eng. Progress, , Vol.86, No.8, pp.55–60.

Bourguignon B., Marcenac F., Keller H.R., de Aguiar P.F. and Massart D.L. (1993): *Simultaneous optimization of pH and organic modifier content of the mobile phase for the separation of chlorophenols using a Doehlert design.* — J. Chromatography, Vol.628, No.2, pp.171–189.

Cooper D.J., Megan L. and Hinde R.F. Jr. (1992): *Comparing two neural networks for pattern based adaptive process control.* — AIChE J., Vol.38, No.1, pp.41–55.

Cotton M.L. and Down G.R.B. (1983): *Reversed-phase high-performance liquid chromatography of sulindac and related compounds using a computer simulation.* — J. Chromatography, Vol.259, No.1, pp.17–36.

Demuth H. and Beale M. (1994): *Neural Network Toolbox for Use with Matlab: User's Guide.* — Natick: MA, The Math Works.

Doehlert D.H. (1970): *Uniform shell designs.* — Applied Statistics, Vol.19, No.2, pp.231–239.

Felinger A. and Guiochon G. (1992): *Optimization of the experimental conditions and the column design parameters in overloaded elution chromatography.* — J. Chromatography, Vol.591, No.1/2, pp.31–45.

Gallant S.R., Kundu A. and Cramer S.M. (1995a): *Optimization of step gradient separations: Consideration of nonlinear adsorption.* — Biotechn. Bioeng., Vol.47, No.3, pp.355–372.

Gallant S.R., Kundu A. and Cramer S.M. (1995b): *Modeling non-linear elution of proteins in ion-exchange chromatography.* — J. Chromatography A, Vol.702, No.1/2, pp.125–142.

Gray R.M. (1984): *Vector quantization.* — IEEE ASSP Magazine, Vol.1, No.2, pp.4–29.

Gu T. (1995): *Mathematical Modeling and Scale-Up of Liquid Chromatography.* — New York: Springer.

Hinde R.F. Jr. and Cooper D.J. (1993): *Using pattern recognition in controller adaptation and performance evaluation.* — Proc. *American Control Conference*, June 2–4, San Francisco, CA, Vol.1, pp.74–78.

Hinde R.F. Jr. and Cooper D.J. (1994): *A pattern-based approach to excitation diagnostics for adaptive process control.* — Chem. Eng. Sci., Vol.49, No.9, pp.1403–1415.

Huo Q. and Chan C. (1995): *Contextual vector quantization for speech recognition with discrete hidden Markov models.* — Pattern Recog., Vol.28, No.4, pp.513–517.

Hu Y. and Massart D.L. (1989): *Uniform shell designs for optimization in reversed-phase liquid chromatography.* — J. Chromatography, Vol.485, No.1, pp.311–323.

Klein, E.J. (1997): *Optimization of protein separations using ion-exchange liquid chromatography.* — Master's Thesis, Stevens Institute of Technology, Hoboken, NJ.

Lindberg W., Johansson E., and Johansson K. (1981): *Application of statistical optimization methods to the separation of morphine, codeine, noscapine and papaverine in reversed-phase ion-pair chromatography.* — J. Chromatography, Vol.211, No.2, pp.201–212.

Linde Y., Buzo A. and Gray R.M. (1980): *An algorithm for vector quantizer design.* — IEEE Trans. Communic., Vol.COM28, No.1, pp.84–95.

Linfield F. and Penny J. (1995): *Numerical Methods Using Matlab.* — New York: Ellis Horwood.

Lundell N. and Markides K. (1993): *Optimization strategy for reversed-phase liquid chromatography of peptides.* — J. Chromatography, Vol.639, No.2, pp.117–127.

Palasota J.A., Leonidou I., Palasota J.M., Chang H. and Deming S.N. (1992): *Sequential simplex optimization in a constrained simplex mixture space in liquid chromatography.* — Anal. Chem. Acta, Vol.270, No.1, pp.101–106.

Press W.H., Flannery B.P., Teukolsky S.A. and Vetterling W.T. (1986): *Numerical Recipes: The Art of Scientific Computing.* — Cambridge: University Press.

Walters F.H., Parker L.R. Jr., Morgan S.L. and Deming S.N. (1991): *Sequential Simplex Optimization: A Technique for Improving Quality and Productivity in Research, Development and Manufacturing.* — Boca Raton: CRC Press Inc.

Wang Q.S., Fie W.Q. and Fan D.P. (1986): *Advanced simplex optimization of two-factor selectivity by high performance thin-layer chromatography.* — Chromatographia, Vol.35, No.3/4, pp.149–152.

Wang Q.S., Gao R.Y. and Yan B.W. (1991): *Computer-assisted optimization of pH and ion concentration selectivity in HPLC using a mixture design simplex method.* — J. Liquid Chromatography, Vol.14, No.16/17, pp.3111–3124.

Yamamoto S., Nakanishi K, Matsuno R. and Kamikubo T. (1983): *Ion exchange cchromatography of proteins — Prediction of elution curves and operating conditions. I. Theoretical considerations.* — Biotechn. Bioeng., Vol.25, No.6, pp.1465–1483.