Curve Fitting Using Natural Computation

A. P. De Weijer, C. B. Lucasius, L. Buydens, and G. Kateman
Department of Analytical Chemistry, Catholic University of Nijmegen, Toernooiveld 1, NL-6525 ED Nijmegen, The Netherlands

H. M. Heuvel and H. Mannee
Akzo Research Laboratories Arnhem, Arnhem, The Netherlands

In curve fitting the most commonly used technique is an iterative hill-climbing procedure that makes use of partial derivatives to calculate the steepest path to an optimum in solution space. However, reliable and accurate initial estimates of the number of peaks, individual peak positions, heights, and widths are necessary to find acceptable solutions. One of the main drawbacks involved is that as the number of overlapping peaks increases, the problem becomes progressively more ill-conditioned. Consequently, small errors in the data (e.g., noise or baseline distortions), errors in the mathematical model, or errors in the estimates can be magnified, leading to large errors in the parameters of the final model. In addition to this, more overlapping peaks can lead to ambiguous fitting results. Ambiguous fitting is a general problem in curve fitting and is not limited to the steepest hill-climbing methods only. In this article we present a method for peak detection using artificial neural networks and a global search technique for curve fitting based on evolutionary search strategies which does not need accurate estimates and is less sensitive to local optima than steepest descent procedures. These statements are corroborated in our comparative case study, which involves the fitting of a series of spectra with strongly overlapping peaks: X-ray equator diffractometer scans of poly(ethylene naphthalate) yarns.

If the underlying mathematical model of the peak pattern is not known, or proper estimates of the parameters in the model assumed cannot be obtained, curve fitting can be a long and complicated task. Apart from this, a good fit of an experimental spectrum does not always lead to parameters with a valid physical meaning because many overlapping peaks may lead to many sets of parameters that can give a close fit of the profile. Vandeginste and De Galan evaluated curve fitting in infrared spectrometry. They investigated the influences of the degree of overlap, the number of unresolved bands in the profile, and the determination of the baseline position on the fitting results of theoretical and experimental spectra. They formulated conditions to be fulfilled in order to obtain reliable results from the fit of infrared data.

Pierce et al. formulated the main sources of errors in curve fitting. These sources are used as a guideline throughout this article: (1) The exact number of peaks is not known, leading to poor convergence or to ambiguous solutions. (2) There is uncertainty about the baseline position. (3) Initial estimates of the parameters in the model are insufficiently accurate. This may cause the problem that the fitting procedure ends in a local optimum.

Error sources 1 and 2 originate from the experimental data, including the way the latter are collected and prepared for the subsequent optimization method used for curve fitting. In contrast, error source 3 originates from the optimization method. Traditionally, local searching optimization methods are most widely applied, e.g., the Gauss–Newton minimization method for nonlinear models. In contrast, globally searching optimization methods are more robust, i.e., less sensitive to initial estimates. Admittedly, globally searching optimization methods are computationally more intensive as well, but with the advent of increasingly faster and cheaper computers, computation times within practical limits become feasible.

In this paper we present the results of our contributions to research concerning these three sources of errors.

THE NEED FOR PEAK DETECTION METHODS

A method is required that allows an estimate of the number of bands and band positions to prevent the problems that were described previously as the first source of error. A high signal-to-noise ratio (SNR) is needed in the original spectra if high-order derivative techniques are used for signal tracing. A popular, advanced method to improve SNR is due to Savitzky and Golay. Jackson and Griffiths compared Fourier self-deconvolution (FSD), a method that constitutes a linear method of deconvolution based on measured data in the time domain. This technique seems to perform better than techniques needing high-order numerical derivatives, but FSD needs estimates of the peak shapes and widths at half-height for proper deconvolution. For the sake of brevity, we use the term “half-width” to denote “full width at half-maximum height” further on. Wyhoff et al. described an artificial neural network for peak verification in noisy infrared spectra. They concluded that a great deal of potential in applying artificial neural networks for peak recognition exists. Recently, a


cerebellar model arithmetic computer (CMAC) neural network was developed for deconvolution of a system of two overlapping chromatographic peaks. This CMAC network is able to provide rapid deconvolutions of simulated and actual chromatographic peaks.6 We present a neural network that estimates the number of peaks, peak position, and half-width of spectra, diffractograms, or chromatograms that are composed of bell-shaped peaks.

Peak Detection Using Artificial Neural Networks. We present an alternative method, based on artificial neural networks (ANN), that is capable of detecting peaks in a spectrum. ANNs are now frequently subjected to calibrate nonlinear relations and to pattern recognition with great success.7-9 ANNs differ from other approaches in that they learn from examples. The ANN algorithm iteratively samples the examples and learns from the mistakes made in previous trials. This process will continue until all examples are mapped in an acceptable way. For further reading, the authors recommend the following literature.10 The ANN discussed here has been trained to estimate the number of bands, peak position(s), and bandwidth(s) of experimental spectra. A series of synthetic patterns of peaks varying from Lorenzian to Gaussian shapes with different degrees of overlap is used to train the ANN. As a mathematical peak model, a sum of several Pearson VII lines was used (eq 1). The Pearson VII function provides the possibility of using a wide variety of line shapes from Gaussian to Lorenzian and beyond.11

\[
A(v) = \sum_{i=1}^{n} A_{i,0} \left[ 1 + 4Z_i^2 \left( 2^{1/m_i} - 1 \right) \right]^{m_i}
\]

where \(A_{i,0}\) is the absorbance in the center of the peak \(i\), \(Z_i = (v - \nu_{i,0})/H_i\) is the peak position of peak \(i\), \(H_i\) is the half-width of peak \(i\), \(m_i\) is the tailing factor of peak \(i\), and \(n\) is the number of peaks.

The ANN, trained to recognize \(\nu_0\) in overlapping peak patterns, uses the first and second derivatives of \(A(v)\) to predict the presence or absence of a peak in \(\nu\). However, the relation between these parameters and the presence or absence of a peak was learned by an ANN and not derived from mathematical formulations.

EXPERIMENTAL SECTION

The source code of the ANN program is written in Turbo Pascal and runs on an 80486 processor under MSDOS. Error back-propagation is used as a learning rule. A multilayered perceptron feed-forward network with four input nodes, one hidden layer with two nodes, and one output node was used for this task. More hidden layers as well as more hidden units did not improve the results. The input nodes represent the following factors. (Node 1) sign change of \(dA(v)/dv\): 0, no sign change; -1, sign change from + to -; 1, sign change from - to +. (Node 2) value of \(d^2A(v)/dv^2\): 1, positive; -1, negative or zero. (Node 3) sign change of \(d^2A(v)/dv^2\): 0, no sign change; -1, sign change from - to +; 1, sign change from + to -. (Node 4) value of \(dA(v)/dv\): 1, positive; -1, negative or zero. (Output node 1) peak present within half-width: 1, \((\nu_{i,0} - 0.5H_i) > \nu > (\nu_{i,0} + 0.5H_i)\); 0, \(\nu\) outside \([\nu_{i,0} - 0.5H_i, \nu_{i,0} + 0.5H_i]\).

The representation of the input pattern allows \(2^4 \times 2^4 = 36\) realization possibilities. Therefore the training set consisted of 36 input patterns and their corresponding binary output patterns. Since the ANN depends on high-order derivatives, the method is sensitive to the SNR. This ANN was trained under ideal circumstances: no baseline distortions or noise was present. For the learning rate (\(\eta\)), a typical value of 0.6 was taken.

Validation. Due to the sigmoid transfer function, the values of the output units of the neural network are usually not exactly equal to 1 or 0. The following arbitrary thresholds were used to force estimation \(H_i\) and \(\nu_{i,0}\) from the neural network output \(O(v)\).

\[
\begin{align*}
\text{if } O(v) < 0.7 & \quad \text{and } O(v + 1) \geq 0.7 \quad \text{then } \nu_s = \nu \\
\text{if } O(v) \geq 0.7 & \quad \text{and } O(v + 1) < 0.7 \quad \text{then } H_i = \nu_b - \nu_s \\
& \quad \text{and } \nu_{i,0} = (\nu_s + \nu_b)/2
\end{align*}
\]

Since this ANN only triggers on sign changes of first and second derivatives, it reacts irrespective of the level and therefore irrespective of the SNR ratio, as is shown as an example in Figure 1. Although the training set was set up using synthetic spectra without noise, the ANN was capable of detecting peaks in noisy synthetic peak patterns (Figure 1) and in experimental peak patterns, provided that some noise was removed; to that end, a simple filter sufficed. To test the reliability more systematically, the performance of this ANN was tested in comparison to peak detection by looking at the numerical second-derivative method only (2ND) and human

---

**Figure 1.** A trained ANN used to detect peaks by scanning the profile. Since this ANN depends on numerical derivatives, it is sensitive to noise. We merely applied a moving average filter to the original profile.
experts (HE) working in the field of applied spectroscopy. Several synthetic peak patterns with different degree of SNR and overlap were analyzed. The validation set consisted of 720 synthetic spectra based on a full factorial design. The factors involved were as follows: resolution (four levels), variance of peak amplitude (four levels), number of peaks (five levels), and SNR (three levels). Each design point was represented three times in the data set. A detected peak was classified as correct when the predicted peak position existed between \[ \nu_{\text{p},0} - 0.5H_n, \nu_{\text{p},0} + 0.5H_n \]. Once a peak has been assigned, other detected peaks in the interval \( [\nu_{\text{p},0} - 0.5H_n, \nu_{\text{p},0} + 0.5H_n] \) were classified as false positives.

Since ANN and 2ND are sensitive to noise, the data set was pretreated by applying a moving average filter of \( n = 3 \) on the original spectra. In Figure 2 the influence of SNR on the percentage of correct classified and false positive peaks for HE, ANN, and 2ND is shown. Irrespective of the filtering, there is an effect of SNR on the number of false positive detected peaks, especially with poor SNR. As stated previously, error source 1 originates from errors in the underlying mathematical model. Therefore it is essential that the number of false positive peaks is as low as possible and that the number of correct peaks is as high as possible. Undoubtedly, the human expert performs better than the ANN and the second-derivative method. However, for practical SNR ranges \( [1, 100] \), this ANN is a reasonable and fast alternative, while the 2ND method exhibits an unacceptable number of false positive peaks. More advanced filter methods may enlarge the SNR range for which ANN peak detection is feasible.

As a "real-world" example, a prediction of the peak position of an experimental X-ray diffractometer scan of poly(ethylene–naphthalene-2,6-dicarboxylate) is shown in Figure 3. To increase the SNR, a moving average block filter was used with a window size of five data points. The diffractometer scans consisted of 470 data points. Note that the shoulder on the left-hand side of the rightmost curve was detected as a peak. The presence of this peak was confirmed by unit cell studies as a \( \beta(200) \) reflex.\(^{12}\) Estimates of \( \ell \) values of the

\( \text{2theta (degrees)} \)

\begin{align*}
5 & \quad 10 \\
15 & \quad 20 \\
25 & \quad 30 \\
35 & \quad 40 \\
45 &
\end{align*}

\text{Figure 2. Scanned profile of an experimental X-ray equator diffractogram.}

\begin{align*}
\text{2theta} & \quad \rightarrow \\
5 & \quad 10 \\
15 & \quad 20 \\
25 & \quad 30 \\
35 & \quad 40 \\
45 &
\end{align*}

\text{Figure 3. SD fit of a PET X-ray equator diffractogram. The fitted curve closely matches the profile.}


**IGNORANCE OF THE BASELINE POSITION**

The severity of baseline detection is strongly related to the number of peaks and the proportion of peak overlap in a spectrum. If there are data points in which no contribution from peaks are present, the baseline position can be fairly well estimated by means of fitting if pure baseline points are situated at several positions in the ordinate of the spectrum. In complex spectra, such as infrared spectra, many overlapping bands often prohibit the estimation of the baseline position. If the number of bands or peaks is not known, it is impossible to determine the baseline position beforehand. A possible strategy is to fit the baseline together with the peak profile. Unfortunately, this can give rise to ambiguous fitting results or ending in local optimal points since baselines strongly interfere with the tailing factors of the peaks. In our experiments done so far, estimates of experimental spectra of X-ray diffractometer scans can be obtained easily since peaks do not interfere with a baseline. There were no diffraction patterns present at low angles \(<9^\circ\) and at high angles \(>30^\circ\) on the equator diffraction patterns. An exponential baseline is assumed to compensate for the scattered light from the unrefracted X-ray beam. Research on more complicated spectra to determine baseline positions, like infrared spectra, will be done in the near future.

**CURVE FITTING WITH STEEPEST DESCENT METHODS**

In our laboratory, a steepest descent based curve-fitting module with the Gauss–Newton optimization method is
Figure 4. (Above) Fraction false positive detected peaks in total of 2880 peaks for human expert (HE), artificial neural network (ANN), and second derivative (2ND). (Below) False negative detected peaks.

Figure 5. SD fits of PEN X-ray equator diffractograms. The scans originate from measured duplicates of the same sample. Although the initial estimates were the same, SD fits ended in different optimal points.

in order to prevent the fitting procedure from ending in a local optimum when SD optimization is used. Moreover, we noticed that more time has to be spent on the development of a generally applicable model as the complexity of the spectra increases. The development of a six-line model of nylon 6 yarns, containing large γ crystals in the presence of α crystals, showed that constraints on form factors and constant ratio of totally diffracted radiation had to be incorporated into the computer model for a correct convergence of the SD optimization. Recent studies on the morphology of poly(ethylene naphthalate) (PEN) showed that curve fitting with SD needs even more accurate initial estimates. It was not possible to formulate constraints to decrease the dimensionality of the optimization problem. In Figure 5, two X-ray equator scans of PEN are shown. These two scans originate from measured duplicates of the same sample, so they only differ in the constitution of the noise. The dashed line represents the fitting with SD of the diffractometer scans with a seven-line model. Six lines were detected by ANN peak tracing over a series of PEN yarns, and one extra line was added to compensate for amorphous scattering. The initial setting parameters are shown in Table 1. The optimization criterion was a minimum residual variance. Both curve-fitting experiments reached convergence. The first sample, however, ended in a fit (in terms of half-widths, peak position, etc.) that was different from that of the second sample, although the initial estimates were the same. This illustrates that small errors in the data caused by noise can be magnified to give large errors in the
parameters of the final model. Since both fits have reached
convergence, we may conclude that the SD ended in two
different local optimal points or that there is a significant
model error which caused ambiguous fitting.

A NEW APPROACH: CURVE FITTING BASED ON
GENETIC ALGORITHMS

It is undesirable to force the optimization method to a
solution by imposing constraints if the problem is actually not
overdimensionalized, i.e., if there is one combination
of parameters that leads to a residual fit in the order of magnitude
of the noise. Such an undesirable decision, to which we were
sensitivity to ending up in local optima, i.e., for their tendency
adhered search methodology which embraces principles of
Genetic algorithms comprise a powerful and increasingly
are robust, i.e., feature a
local optimal points has been developed.

This procedure is based on so-called genetic algorithms. Genetic algorithms comprise a powerful and increasingly
adhered search methodology which embraces principles of
Darwinian evolution. They are especially suitable for complex,
large-scale optimization; for more details about their concepts
and operation, the reader is referred to refs. 18-22

Genetic algorithms are generally praised for their insen-
tivity to ending up in local optima, i.e., for their tendency
to approach the globally optimal solution irrespective of diverse
starting conditions. We therefore say that genetic algorithms
are robust, i.e., feature a good search accuracy. By contrast,
the search precision of genetic algorithms is poor: Upon
replicating runs, there is a considerable spread in the end
solutions, despite the good search accuracy; the average end
solution is a reliable estimate of the global solution in general.

Importantly, opposite properties often apply to traditional
local optimization techniques, such as the aforementioned steepest descent method: a good search precision and a poor
search accuracy, as pointed out above. Therefore, better
overall performance can be attained in a sequential combi-
nation, where the genetic algorithm generates a “best guess”
that serves as a starting point for subsequent improvement
(refinement) by the local technique. Such (and other) hybrid
techniques are important in that the constituent methods
mutually supplement each other for enhanced overall per-
formance.

Hybridization for Enhanced Search. We consider three
basic strategies to hybridization of genetic algorithms: pre-
hybridization, posthybridization, and self-hybridization.

Prehybridization is concerned with finding an initial estimate for a genetic algorithm: in our case, initial values for the Pearson VII model parameters. Even when this estimate is considerably inaccurate, which is mostly the case, the genetic algorithm generally still performs well, as we will show. In our application, the rough initial estimate is obtained from the aforementioned ANN for peak detection. Technically, an initial estimate for a genetic algorithm is called a
working point. It forms the center of the real space in which
the genetic algorithm searches. This space is also called the
search volume, and its dimensions are specified by the user.
The value ranges for the model parameters thus define the
dimensions of the search volume in our application, i.e., their
low bounds and high bounds. For each model parameter,
only a limited number of values, or levels, specified by the
user, is considered by a genetic algorithm. For convenience,
these levels are chosen equidistant. They amount to a search
grid in the search volume. The mesh sizes that characterize
the search grid dictate the search precision that can be attained
by the genetic algorithm concerned. Randomly selected nodes
on the search grid comprise the starting point of the search.

Posthybridization is concerned with the improvement
(refinement) of the end solution found by a genetic algorithm.
To that end, a local search method is used; e.g., we used the
steepest descent method in our application. In this way, the
local search comprises a “remedy” for the poor search precision
as a shortcoming of a genetic algorithm.

Self-hybridization is a strategy wherein a genetic algorithm
is pre- or posthybridized with another. We implemented this
approach as a chain of genetic algorithms wherein each passes
its result as the working point of the next in the chain. Thereby,
in each step the search grid was kept invariant and the search
volume was pruned (Figure 6) in order to accomplish
significant reductions in overall convergence time, to gain
search precision, and to preserve good search accuracy
simultaneously. The first few genetic algorithms in the chain
are deliberately not run up until convergence, as this is not
needed for the next.

Representation and Search Heuristics. Any search tech-
nique produces candidate solutions in an iterative way. These
are guesses at the true solution of the problem concerned. A
candidate solution is represented as a string (vector) of
proposed values for the unknown parameters. Any string is
evaluated by an objective function for assessment of its quality,
i.e., of its likeness to represent the true solution. A string
may be modified by the search heuristics in an attempt to
represent a better candidate solution.

A distinguishing feature of genetic algorithms is that their
evolutionary search heuristics manipulate a population of
strings. In each iteration, or generation, the population
is replaced by a new population of equal size. The new population

---

Table 1. Initial Values for Each Parameter in a Seven-Line
PEARSON VII Model Obtained after Prehybridization for SD

<table>
<thead>
<tr>
<th>Peak</th>
<th>(A_0)</th>
<th>(b_0)</th>
<th>(H)</th>
<th>(2 - (1/m))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.04</td>
<td>13.47</td>
<td>0.86</td>
<td>1.98</td>
</tr>
<tr>
<td>2</td>
<td>0.10</td>
<td>15.62</td>
<td>1.08</td>
<td>1.38</td>
</tr>
<tr>
<td>3</td>
<td>0.80</td>
<td>18.68</td>
<td>1.41</td>
<td>1.61</td>
</tr>
<tr>
<td>4</td>
<td>0.08</td>
<td>23.97</td>
<td>1.99</td>
<td>1.69</td>
</tr>
<tr>
<td>5</td>
<td>0.88</td>
<td>26.20</td>
<td>0.85</td>
<td>1.82</td>
</tr>
<tr>
<td>6</td>
<td>0.20</td>
<td>26.50</td>
<td>1.41</td>
<td>1.17</td>
</tr>
<tr>
<td>7</td>
<td>0.10</td>
<td>20.50</td>
<td>11.0</td>
<td>1.98</td>
</tr>
</tbody>
</table>

(17) van den Heuvel, C. J. M.; Heuvel, H. M.; Faassen, W. A.; Veurink, J.; Lucas,
L. J. Molecular changes of PET yams during stretching measured with reo-
49, 925-934.


15, 1-33.

(20) Lucasius, C. B.; Katenman, G. Understanding and using genetic algorithms.
Part 2: Representation, configuration and hybridization, Chemom. Intell.
Lab. Syst., in press.

(21) Lucasius, C. B.; Katenman, G. GATES: genetic algorithm toolbox for
evolutionary search. Software library in ANSI C, Laboratory for Analytical
Chemistry, Catholic University of Nijmegen, January 1991.

(22) Lucasius, C. B.; Katenman, G. GATES towards evolutionary large-scale
optimization: a software-oriented approach to genetic algorithms. (a) Part 1:
subdivided into $2^B$ levels; thus, larger $B$ values amount to a finer meshed search grid.

Many genetic operators exist for the modification of binary strings. We used B-UX (uniform binary recombination) and B.M (uniform binary mutation). B-UX is applied with probability $P_r$ to the twosomes of bitstrings obtained after randomly pairing the strings in the population; for each successful trial, it swaps—positionwise and with probability $P_s$—bits between both strings that make the pair; if brief, B-UX is parameterized by $Pr$ (recombination probability) and $Ps$ (swap probability); important advantages of B-UX are maximum exploratory power and positional unbiasedness.\textsuperscript{22} B.M inverts the bits in the population with probability $P_m$; in brief, B.M is parameterized by $P_m$ (mutation probability).

**Objective Function.** The evaluation of a bitstring proceeds as follows. First, the bitstring is decoded into a string of real values for the unknown parameters; a detailed description of this procedure can be found in ref \textsuperscript{22}. The real values are then passed to the objective function. It calculates a spectrum according to the Pearson VII model and subsequently compares this spectrum with the known experimental spectrum to derive a measure of (dis)similarity. Two criteria that seem sensible are root mean square (RMS, as dissimilarity) and correlation coefficient (CORR, as similarity). Both criteria have shortcomings, though. For instance, CORR is not fully consistent with the purposes of curve fitting because it quantifies ratios rather than differences between spectra. RMS, on the other hand, is consistent but tends toward "plateau" optima on the RMS landscape; this follows from observed "indecisive" search behavior near optima and may also be appreciated intuitively by considering small lateral perturbations in a spectrum. In order to obtain a criterion that is both consistent and leads to a structurally more pronounced landscape, we defined the bicriterion $\text{ERROR} = \text{RMS}/(1 + \text{CORR})^2$ (dissimilarity). Experimentally, minimization of ERROR led to a significantly better performance of our genetic algorithm.

**Configuration.** The configuration is specified in a separate input file. For reasons of limited space, we do not provide a transcript of this file here, but merely summarize its key entries as follows:

<table>
<thead>
<tr>
<th>peak</th>
<th>$A_0$</th>
<th>$\theta_0$</th>
<th>$H$</th>
<th>$2 - (1/m)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1_\beta(100)$ low bound</td>
<td>0.00</td>
<td>12.63</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>$1_\beta(100)$ high bound</td>
<td>0.13</td>
<td>14.33</td>
<td>2.71</td>
<td>1.98</td>
</tr>
<tr>
<td>$2_\alpha(010)$ low bound</td>
<td>0.00</td>
<td>14.63</td>
<td>0.17</td>
<td>1.00</td>
</tr>
<tr>
<td>$2_\alpha(010)$ high bound</td>
<td>0.25</td>
<td>16.83</td>
<td>1.87</td>
<td>1.98</td>
</tr>
<tr>
<td>$3_\beta(100)$ low bound</td>
<td>0.21</td>
<td>17.82</td>
<td>0.54</td>
<td>1.00</td>
</tr>
<tr>
<td>$3_\beta(100)$ high bound</td>
<td>0.61</td>
<td>19.52</td>
<td>2.24</td>
<td>1.98</td>
</tr>
<tr>
<td>$4_\alpha(100)$ low bound</td>
<td>0.00</td>
<td>23.29</td>
<td>1.39</td>
<td>1.00</td>
</tr>
<tr>
<td>$4_\alpha(100)$ high bound</td>
<td>0.33</td>
<td>24.14</td>
<td>3.09</td>
<td>1.98</td>
</tr>
<tr>
<td>$5_\beta(100)$ low bound</td>
<td>0.57</td>
<td>25.52</td>
<td>0.33</td>
<td>1.00</td>
</tr>
<tr>
<td>$5_\beta(100)$ high bound</td>
<td>1.17</td>
<td>26.88</td>
<td>1.52</td>
<td>1.98</td>
</tr>
<tr>
<td>$6_\alpha(100)$ low bound</td>
<td>0.00</td>
<td>26.45</td>
<td>1.10</td>
<td>1.00</td>
</tr>
<tr>
<td>$6_\alpha(100)$ high bound</td>
<td>0.55</td>
<td>27.06</td>
<td>1.86</td>
<td>1.98</td>
</tr>
<tr>
<td>7 amorph low bound</td>
<td>0.00</td>
<td>20.30</td>
<td>4.25</td>
<td>1.98</td>
</tr>
<tr>
<td>7 amorph high bound</td>
<td>0.15</td>
<td>23.70</td>
<td>12.8</td>
<td>1.98</td>
</tr>
</tbody>
</table>
For a more detailed explanation of terminology related to configuration, the reader is referred to ref 22.

**Software and Hardware.** The routines that comprise a genetic algorithm consist of two segments: a domain-dependent part (concerning the problem representation and the objective function) and a domain-independent part (concerning of evolutionary search heuristics). By "domain-independent" routines are meant routines that can be used for other domains as well; for our application these were obtained from the software library GATES.21,22 In agreement with this library, the domain-dependent routines and data structures were programmed in C (ANSI standard) for speed and portability. The integration of all routines resulted in CFIT: the executable application program for genetic curve fitting.23 CFIT is presently available for MSDOS and UNIX systems. The CFIT fitting procedure used in this application was executed under MSDOS with an 80486 processor.

**COMPARISON OF FITTING RESULTS GA AND SD**

The same two diffractograms that were fitted with the SD procedure were now fitted with CFIT, using the same seven-line mathematical model. Prehybridization on the original data to determine the search volumes was carried out as described in the previous section. The low and high bounds for each parameter are given in Table 2. The dimensions of the initial search volume were too small for effective self-hybridization. With the ANN peak detection procedure it was possible to reduce the search volume in such a way that self-hybridization was not useful. We expect that the ANN preprocessing only for reducing the search volume will not be enough for curve fitting of infrared spectra. Refinement of the end solution found by the genetic algorithm by a steepest descent method did not reduce the residual variance of the sums of squares significantly, so posthybridization was not necessary. In Figure 7, the fitting results of a genetic algorithm are shown for both experimental diffractograms. Since the residual variance is of the order of magnitude of the experimental noise, we may conclude that the Gauss–Newton steepest descent approach (Figure 5) ended in a local optimal point and that the possibility of a significant model error can be ruled out. For the poorly resolved peaks 1 and 6, the SD gave significantly different solutions for the half-widths. The standard deviations were calculated from the generalized inverse of the Jacobian matrix multiplied by the residual variance (see Appendix).24 Curve fitting with GA showed compatible values for all half-widths in both diffractograms (Figure 8). So the GA approach is independent of the noise in this special case. To shown that this is also true of a more general case, the following experiment has been set up.

A series of PEN yarns has been produced resulting from a variation of one process parameter $\xi$ in 10 discrete levels. For all these yarns X-ray equator diffractograms were recorded. These scans were all fitted with the seven-line model with SD and GA, as described previously. As an illustration of the fitting results, the half-width of the first poorly resolved peak is plotted against the process parameter $\xi$ for SD and GA (Figure 9). Since this is a univariate relation, a smooth relation between $\xi$ and half-width is expected. The determination of half-widths using SD showed severe scattering, however. Although the variance of each separate fit was rather low, it was not possible to follow the half-width of the first peak as a function of $\xi$, due to the sensitivity of SD to local optima. As expected, the GA performed better.

In Table 3, some properties of three optimization procedures, namely, steepest descent (SD), genetic algorithms (GA), and exhaustive search (ES), for this seven-line diffractogram of PEN are shown. Any optimization technique applied to ill-conditioned problems has the disadvantage that there is no certainty about convergence into a global optimum. Since the calculation time for exhaustive search is exponentially proportional to the number of peak parameters, it therefore exceeds reasonable time limits in most situations. In general,

![Figure 7. GA fit of X-ray equator profiles of PEN. Both fits closely match the diffractograms.](image)

---


---

*Analytical Chemistry, Vol. 66, No. 1, January 1, 1994*
Figure 8. Comparison of estimation of half-widths at half-height (H) of two repeated measurements with GA and SD. Peaks which were not very well separated gave different results with SD and identical results with the new GA method.

Figure 9. Estimation of half-width of the β(100) reflex in a univariate relation with SD and GA curve fit method.

Table 3. Comparison of Genetic Fit (GA), Steepest Descent Fit (SD), and Exhaustive Search (ES) for This Application

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>CPU time (min)</th>
<th>optimum type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES</td>
<td>minimal</td>
<td>minimal</td>
<td>~10^6</td>
<td>global</td>
</tr>
<tr>
<td>GA</td>
<td>0.92</td>
<td>0.89</td>
<td>unknown</td>
<td>35</td>
</tr>
<tr>
<td>SD</td>
<td>9.1</td>
<td>6.4</td>
<td>yes</td>
<td>2</td>
</tr>
</tbody>
</table>

The SD was only 2 min per run under MSDOS, but it should be added that at least 30 different runs were performed to achieved this best-ever solution, which was still unacceptable.

**FINAL REMARKS AND OUTLOOK**

In fitting complex spectra, in which many bands display large overlaps, the steepest descent approach can fail because very accurate initial estimates of the fit parameters have to be given to optimize these initial estimates to the global optimum. We have shown that genetic algorithms are less sensitive to local optima in our experiments in which we successfully revealed underlying peaks in highly overlapping peak patterns. Our research is now aimed at fitting infrared spectra using genetic algorithms. The steepest descent method has been successfully applied to the fitting of poly(ethylene terephthalate) (PET) yarns, but worked only under highly constrained conditions. Many constraints were imposed without any acceptable physical background and were only used to force the optimization to convergence. The formulation of the constraints was a time-consuming task. Approximately 100 band parameters are involved in the fitting of infrared spectra of PET. Our results indicate that genetic algorithms are useful or even essential in the determination of band parameters of such a high-dimensional problem within reasonable time.

**ACKNOWLEDGMENT**

The authors acknowledge Akzo Research Laboratories Arnhem for their financial support to this work. Furthermore, we thank D. Nijland for the fruitful discussions concerning neural network testing and C. J. M. van den Heuvel for providing the X-ray diffractograms of PEN.

**APPENDIX**

Estimation of the unknown parameters \( \theta = (\theta_1, \theta_2, ..., \theta_n) \) in a mathematical model of the form

\[
y = f(x; \theta)
\]

where \( y \) is the dependent variable or response and \( x \) is the independent variable or factor. The model is nonlinear in the parameters. The parameters must be bound on both sides by constants:

\[
a_j \leq \theta_j \leq b_j, \quad j = 1, 2, ..., n
\]

The parameter estimates are obtained on the basis of a least-squares fit to \( m \) data points \( (x_i, y_i) \); i.e., a set of parameters \( \hat{\theta} \) is determined such that the sum of squares \( F(\theta) \) is minimal:

\[
F(\hat{\theta}) = \min_{\theta} \sum_{i=1}^{m} (y_i - y(x_i; \theta))^2
\]

This method is based on Gauss–Newton for minimizing
functions of the type extended with a Marquards technique for forcing convergence and a technique for handling constraints. It is an iterative process which starts with a feasible initial estimate between the boundary constants:

$$\theta^0 = (\theta_1, \theta_2, ..., \theta_n)$$

A sequence of new estimates is determined iteratively such that the sum of squares diminishes

$$F(\theta^{k+1}) < F(\theta^k), \quad k = 0, 1, 2, ...$$

This process stops by convergency if at least one of the two following conditions is satisfied: (1) Both the parameters and the sum of squares have been determined with a given accuracy.

(2) The sum of squares does not change significantly (in connection with the relative machine precision).

The variance/covariance matrix of the model parameter is calculated from the Jacobian matrix as follows:

$$(J^TJ)^{-1}s_\tau^2$$

where $s_\tau = SS_{\text{res}}/(n - p)$, $n$ is the number of data points, and $p$ is the number of model parameters.