Available online at www.sciencedirect.com
SCIENCE DIRECT
Talanta 68 (2006) 1336–1342
Talanta
www.elsevier.com/locate/talanta

A hybrid genetic algorithm for estimating the equilibrium potential of an ion-selective electrode

Peter Watkins [a,*], Graeme Puxty [b]

[a] *CSIRO Food Science Australia, Private Bag 16, Werribee 3030, Australia*
[b] *Institute for Chemistry and Bioengineering, Safety and Environmental Technology Group, ETH-Hönggerberg HCI 130, CH-8093 Zurich, Switzerland*

Received 28 April 2005; received in revised form 25 July 2005; accepted 25 July 2005
Available online 29 August 2005

Abstract

Non-linear equations can be used to model the measured potential of ion-selective electrodes (ISEs) as a function of time. This can be done by using non-linear least squares regression to fit parameters of non-linear equations to an ISE response curve. In iterative non-linear least squares regression (which can be considered as local optimisers), the determination of starting parameter estimates that yield convergence to the global optimum can be difficult. Starting values away from the global optimum can lead to either abortive divergence or convergence to a local optimum. To address this issue, a global optimisation technique was used to find initial parameter estimates near the global optimum for subsequent further refinement to the absolute optimum. A genetic algorithm has been applied to two non-linear equations relating the measured potential from selected ISEs to time. The parameter estimates found from the genetic algorithm were used as starting values for non-linear least squares regression, and subsequent refinement to the absolute optimum. This approach was successfully used for both expressions with measured data from three different ISEs; namely, calcium, chloride and lead ISEs.

Keywords: Ion-selective electrodes; Genetic algorithms; Non-linear regression; Equilibrium potential; Estimation; Hybrid

1. Introduction

Measurements with ion-selective electrodes (ISEs) are generally made by allowing the equilibrium potential (E_{eq}) to be established. In some cases, the time required to do so can be quite lengthy. An alternative approach has been to use an extrapolation procedure to estimate E_{eq}. This can be done by sampling a small portion of the response curve and applying linear regression to transformations of the empirical relationships between the measured potential and time [1–6]. One such empirical model was first reported by Buffle and coworkers [7,8] and relates the measured potential, E, with time, t, by

$$E = E_{eq} + 1/(At + B) \quad (1)$$

where A and B are the empirical constants.

While the use of linear least squares regression on transformed linear equations is simple and straightforward, it may not be statistically valid since the original error structures can be altered, which can introduce inaccuracies into the parameter estimates [9]. This can be overcome by using non-linear regression with the original equations such as Eq. (1). This approach has been done for estimating the equilibrium potential for a solid-state lead ISE [10].

In addition to Eq. (1), another empirical model that can be used is

$$E = E_{eq} + de^{-kt} \quad (2)$$

where d and k are empirical constants. This model was first suggested by exploratory data analysis of the response curve for the lead ISE [10]. Non-linear regression though requires reliable estimation of the initial parameters. Otherwise, convergence to the global optimum may not be achieved. Genetic algorithms are capable of providing initial

* Corresponding author. Tel.: +613 97313467; fax: +613 97313250.
E-mail address: Peter.Watkins@csiro.au (P. Watkins).

0039-9140/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.talanta.2005.07.054

parameter estimates [11] that, with further refinement by non-linear regression, can be used to model the equilibrium potential for an ISE. GAs have been used to find initial parameter estimates for non-linear regression of immittance data [12], and to fit non-linear models for electrochemical impedance spectroscopy [13]. We report on the use of this approach for the calcium, chloride and lead ISEs.

1.1. Genetic algorithms

The application of genetic algorithms (GAs) to problems in chemistry and science has only recently been undertaken. Interest in evolutionary algorithms began with the work of Rechenberg [14] and Holland [15] but their early application was limited due to the fact that GAs are computationally intensive. Often thousands, if not millions, of evaluations of the objective function are required to define the fitness of members of the GA. It is only with recent advances in computing power that the use of GAs has become feasible.

Genetic or evolutionary algorithms are a class of adaptive optimisation techniques based on Darwinian principles of natural selection and survival of the fittest. Candidate solutions in GAs, in this case estimates for the parameters of Eq. (1) or (2), need to be encoded as strings of numbers, namely a Gray coded bit string; many other alternate binary or decimal codings are also possible. The fitness, or goodness of fit, for a set of parameters is estimated as the sum of the squares of the residuals; i.e. the sum of the squares of the difference between the measured and calculated ISE data. Thus, the lower the sum of the squares the better the fit of the solution. A set of candidate solutions is called a population and the i-th population within the genetic algorithm will be represented as P_i.

A flow diagram outlining the details of the implemented genetic algorithm is given in Fig. 1. A more detailed description of the implementation has been given elsewhere [16] in regard to the application of the GA to the determination of kinetic model parameters. In this case, the implementation is identical except that the objective function is either Eq. (1) or (2).

In summary, the steps taken in the genetic algorithm are as follows:

Setup: Load the ISE data and define the equation. Define the genetic algorithm's operational parameters (parameter

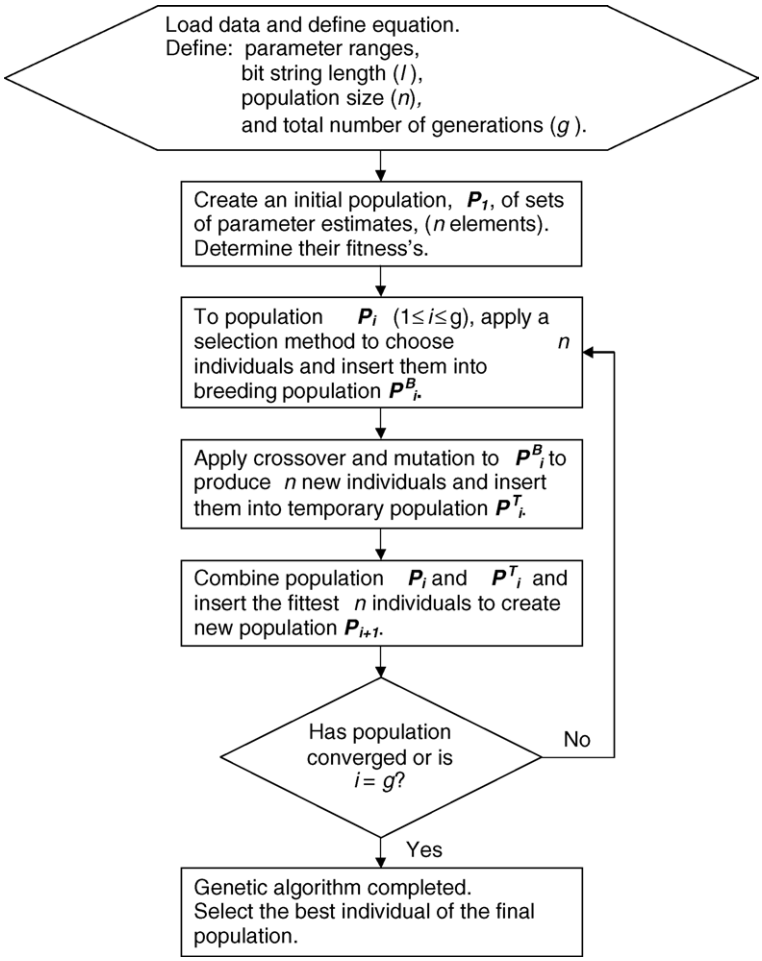

Fig. 1. Flow diagram of the genetic algorithm.

ranges, bit string length (I), population size (n), maximum number of generations (g) and mutation rate).

Initialisaton: Create the first generation P_1 of the genetic algorithm by randomly generating a set of n candidate solutions within their defined range.

Selection: Apply a selection method to the current generation to select candidate solutions for inclusion into a breeding population. There are many different strategies for this selection process and they all need to ensure that the fitter candidates are more likely to be included into the breeding population, but still allow exploration of large areas of the parameter space. In this case the elitist selection method was chosen: n individuals are selected for inclusion in the breeding population P_i^{B}. An individual may be selected multiple times. First, the fittest half of P_i is inserted into P_i^{B}. The remaining $n/2$ spaces in P_i^{B} are filled by tournament selection, where two individuals are randomly selected from the original, complete P_i. Seventy-five percent of the time the fitter individual is placed in P_i^{B}, and 25% of the time the less fit individual is chosen. These procedures enforce a strong preference for the 'good' members of the population, but it also allows 'bad' members to 'survive'.

Crossover and mutation: Crossover is applied to the breeding population. It involves first randomly selecting two candidate solutions from the breeding population. The strings of these candidate solutions are then combined in some way to produce a new candidate solution that retains aspects of both parent strings. The type of crossover implemented was half-uniform crossover where the bits at each location of the selected pair of strings are compared. If the bit at a particular location is the same in both strings it is kept intact. If the bit differs a new bit is taken from either string with equal probability. This is illustrated in Fig. 2.

Mutation is then applied to each new solution resulting from crossover. Each element within the new solution string may randomly change with some given probability (usually less than 10%). Mutation was implemented so that it decreases linearly with the number of generations. Mutation helps prevent premature convergence of the algorithm by providing random disruption so different regions of the solution space will be explored.

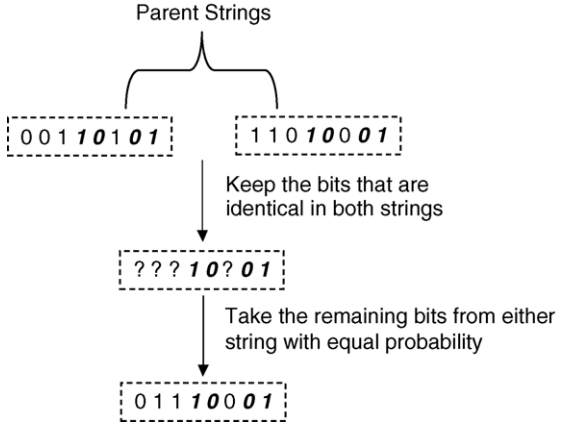

Fig. 2. Half-uniform crossover.

Replacement: The process of crossover and mutation is repeated until the required number n of new candidate solutions has been created and the fitness of each new solution is evaluated. The new solutions are then combined with the current generation producing a population of $2n$. The fittest n solutions are kept and make up the next generation, the rest are discarded. The process of selection through to replacement is then applied to the new generation and so on.

Termination: The genetic algorithm terminates if the maximum number of generations is reached or it has converged to a particular solution.

2. Experimental

Lead nitrate, calcium nitrate and potassium chloride were used for standard preparation and were of analytical grade. The standard solutions were prepared with deionised water. The lead measurements were made with an Orion lead solid-state (94–82) ion-selective electrode with a double junction sleeve type electrode (90–02) as the reference. The ionic strength of the lead standard solutions was adjusted with 5 M ammonium nitrate. The cell potential was measured with an Orion EA940 ion analyser at 0.25 min intervals by hand. The chloride measurements were made with an Orion chloride combination electrode (9617BN) with an Orion 420A meter. The ionic strength of the chloride standard solutions was adjusted with 5 M sodium nitrate. The calcium measurements were made with a Radiometer calcium (ISE-K-CA) ISE interfaced to a PHM93 meter with a double junction sleeve type electrode (Radiometer REF251) as the reference. The ionic strength of the calcium standard solutions was adjusted with 5 M potassium chloride. Ammonium nitrate (10%) solution was used in the outer chamber of the reference electrodes. Measurements were made according to the manufacturer's instructions at room temperature and without stirring. Between each measurement, the electrodes were rinsed with distilled water and blotted dry with a tissue. The data was collated in an Excel spreadsheet and estimates for the non-linear parameters were calculated using the genetic algorithm. An outline of the algorithm is given in Fig. 1. The GA was coded using C++ and details of the implementation can be found elsewhere [16]. The calculated values from the genetic algorithm were used as starting estimates for non-linear least square regression of Eqs. (1) and (2). Non-linear regression was performed using the "nls" command with R, an open-source implementation of the S statistical language [17].

3. Results and discussion

The GA was run three times to calculate estimates for the parameters of Eqs. (1) and (2) with response data for a lead

Table 1
Calculated parameter estimates by genetic algorithm, non-linear regression and bootstrap simulation for Eqs. (1) and (2) for a lead ion-selective electrode

pPb	Genetic algorithm			Iterative non-linear regression			Iteration	Bootstrap[b]		
	E_{CB}	A	B	E_{CB}	A[a]	B		E_{CB}	A	B
Eq. (1)										
2	−139.3	0.086	0.3846	−139.2 ± 0.1	0.01 ± 0.01	0.40 ± 0.01	4	−139.13 ± 0.11	0.13 ± 0.06	0.38 ± 0.17
3	−163.8	6.12–6.36	0.002*	−166.3 ± 0.1	0.05 ± 0.01	0.30 ± 0.01	13	−165.64 ± 0.08	0.11 ± 0.02	0.32 ± 0.02
4	−192.6	7.43	0.002-0.005	−193.8 ± 0.1	0.18 ± 0.02	0.44 ± 0.01	11	−194.09 ± 0.22	0.10 ± 0.05	0.47 ± 0.04
pPb	Genetic algorithm			Iterative non-linear regression			Iteration	Bootstrap[b]		
	E_{CE}	d	k	E_{CE}	d	k		E_{CE}	d	k
Eq. (2)										
2	−139.3	2.39	0.13	−138.6 ± 0.03	1.84 ± 0.03	0.27 ± 0.02	9	−138.71 ± 0.34	1.85 ± 0.34	0.23 ± 0.04
3	−163.8	1.19–1.27	1.22–1.27	−165.2 ± 0.05	2.31 ± 0.04	0.19 ± 0.01	6	−165.16 ± 0.07	2.23 ± 0.04	0.24 ± 0.04
4	−192.6	1.0–27.0	2.24–32.64	−193.5 ± 0.05	1.69 ± 0.05	0.34 ± 0.03	6	−193.74 ± 0.26	1.56 ± 0.19	0.17 ± 0.08

[a] ±Calculated standard error.
[b] Bootstrap estimates taken from [10].

ISE immersed in standard solutions. Table 1 shows the range of the parameter estimates obtained from the GA. Note a single entry in Table 1 for the GA value means that the same result was found with each run. These estimates were then used as starting values for iterative non-linear least squares regression. The calculated non-linear least square parameter estimates for Eqs. (1) and (2) are also shown in Table 1. Note that the calculated E_{eq} is denoted as E_{CB} for Eq. (1) and E_{CE} for Eq. (2). For Eq. (1), the GA parameter estimates are well within a factor of those found with non-linear least squares regression. The parameter estimates from bootstrap simulations are also shown in Table 1. Bootstrap simulations were used to determine the standard error estimates for the non-linear least square parameters for Eq. (1). Bootstrap methods are computer intensive statistical tools that provide uncertainties for the parameter estimates [18,19] by random sampling of the experimental data, over a large number of times. The non-linear least square parameter estimates are in close agreement with those from the bootstrapping simulations. The number of iterations required to reach convergence is reasonably modest where the maximum number of iterations was 13 for the 10^{-3} M standard solution. Thus, the use of the GA values as starting guesses to the iterative algorithm provided good non-linear parameter estimates that converged quite quickly. The same result is found for Eq. (2) where the GA parameter estimates are reasonably close to those calculated with non-linear regression. Additionally, the iterative algorithm converged more rapidly to the global optimum where the maximum number of iterations required was 9. As with Eq. (1), the final non-linear least square estimates also compare very favourably with the bootstrap simulated data. Fig. 3 shows the plot of expected values from each equation with the experimental data for the lead ISE immersed in a 10^{-4} M standard solution. The expected values were calculated using the parameter estimates shown in Table 1. A good fit of the response curve was found with both Eqs. (1) and (2) with only a difference of 0.1 mV between each predicted curve.

The genetic algorithm was applied to response data from calcium and chloride immersed in standard solutions (ISEs) in order to find parameter estimates for Eqs. (1) and (2). Table 2 shows the parameter estimates found by the GA and those calculated by non-linear least squares regression for the calcium ISE. Bootstrap simulated estimations were only performed for Eq. (1) parameter estimates. As with the lead ISE, the GA estimates for Eq. (1) for the calcium ISE are well within a decade of order with the non-linear least square estimates and those found from the bootstrap simulations. Similar agreement is also found for the estimates for Eq. (2). The results for the chloride ISE are shown in Table 3. Except for the pCl 2.7 data, the GA estimates for Eq. (1) compare quite well to the non-linear least square estimates. This is also the case for the parameter estimates of Eq. (2). In contrast, there are differences between the GA estimates and those found by non-linear regression for the pCl 2.7 data. The standard errors for the parameter estimates of Eqs. (1) and (2) for this data are also quite large. Inspection of the response curve (not shown) showed that the electrode had reached the equilibrium potential quite quickly and had not responded

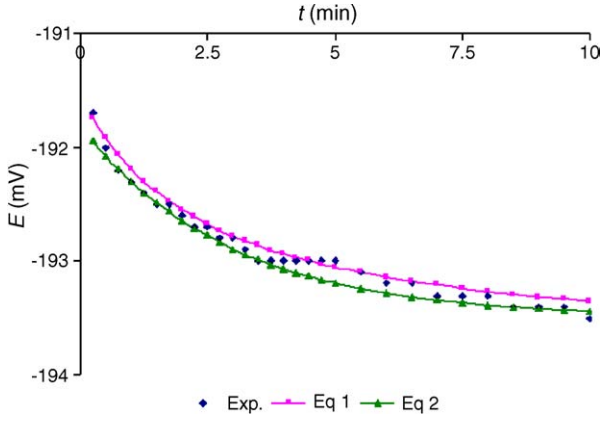

Fig. 3. Plot of expected values from Eqs. (1) and (2) with the measured potential for a lead ISE immersed in 10^{-4} M standard solution.

Table 2
Calculated parameter estimates by genetic algorithm, non-linear regression and bootstrap simulation for Eqs. (1) and (2) for a calcium ion-selective electrode

pCa	Genetic algorithm			Iterative non-linear regression			Bootstrap		
	E_{CB}	A	B	E_{CB}	A	B	E_{CB}	A	B
Eq. (1)									
2	118.5	0.22–0.26	0.002	120.2 ± 0.1	2.29 ± 0.15	−0.50 ± 0.04	120.4 ± 0.1	2.83 ± 0.07	−0.63 ± 0.17
3	147.0	0.1–0.11	0.21–0.22	147.5 ± 0.1	0.21 ± 0.01	0.21 ± 0.01	147.6 ± 0.3	0.22 ± 0.06	0.20 ± 0.01
4	172.9	0.030–0.036	0.15–0.16	175.0 ± 0.2	0.17 ± 0.03	0.17 ± 0.01	174.9 ± 0.7	0.13 ± 0.05	0.12 ± 0.01

pCa	Genetic algorithm			Iterative non-linear regression		
	E_{CE}	d	k	E_{CE}	d	k
Eq. (2)						
2	121.7–121.8	421.3–873.3	12.5–15.9	120.4 ± 0.1	289.0 ± 46.8	10.8 ± 0.8
3	147.0	0.37–3.45	0.16–0.17	148.1 ± 0.1	3.61 ± 0.13	0.53 ± 0.04
4	168.3–177.6	4.91–9.89	0.04–1.83	175.7 ± 0.1	4.46 ± 0.25	0.59 ± 0.06

in a way such as shown in Fig. 3. Thus, the likely cause for this seemingly poor result would be due to a lack of fit of the equations to the pCl 2.7 data. In addition, there are also differences between the calculated E_{CB} and E_{CE} values for the pCl 3.2 data (approximately 10 mV). The reason for this is unclear. Nevertheless, the GA provided very good initial parameter estimates that were used for non-linear regression of Eqs. (1) and (2) applied to the response data of three different ISEs. It is likely that this approach could be used with other ISEs but this would need to be verified.

Fig. 4 shows the performance of the genetic algorithm for three independent runs using the response data of the calcium ISE immersed in 10^{-3} M standard solution. The genetic algorithm was run using the following settings:

- parameter range of 0.001–1000 (with a logarithmic scale),
- generation size of $n = 50$ with a maximum of $g = 100$ generations,
- encoding of $l = 9$ bits resolution per parameter,
- mutation rate of 5%.

Plots are given for Eqs. (1) and (2). The plots show the sum of squares (or 'fitness') of the fittest member of each generation, the mean sum of squares for each generation and the sum of squares of the absolute optimum. It is worth noting that the sum of squares of the absolute optimum can never be achieved by the GA due to the discretisation of the parameter values to allow their representation as a binary string. The plots show the GA performed well with convergence to a near optimal set of parameter estimates within the defined maximum number of generations. It can also be seen that the algorithm performed slightly differently with each run. This highlights the probabilistic nature of the approach.

The maximum run time for the GA was approximately two minutes on a 3 GHz Pentium 4 processor. This is significantly more then a single run for the local optimiser, due to the large number of evaluations of the objective function. However, the benefit of using the GA becomes clear when consideration of the quality of the initial guesses provided to the local optimiser is taken into account. Much more time then a few minutes can be spent when manually trying to determine initial parameter estimates for the local optimiser that result in successful convergence to the global optimum. For example, using initial parameter estimates of $E_{eq} = -192.6$, $d = 0.1$ (one-tenth of the GA value) and $k = 2.24$ for Eq. (2), applied to the 10^{-4} M Pb ISE response data, results in failure of the

Table 3
Calculated parameter estimates by genetic algorithm and non-linear regression for Eqs. (1) and (2) for a chloride ion-selective electrode

pCl	Genetic algorithm			Iterative non-linear regression		
	E_{CB}	A	B	E_{CB}	A	B
Eq. (1)						
3.7	132.0	−0.16 to −0.25	−0.01 to −0.01	132.2 ± 0.25	−0.21 ± 0.02	−0.013 ± 0.01
3.2	75.0	−0.039	−0.013	90.4 ± 1.5	−0.011 ± 0.001	−0.015 ± 0.01
2.7	93.5	2.02 to 3.45	−0.01 to −0.44	93.5 ± 0.1	85.1 ± 598	−20.65 ± 149.5

pCl	Genetic algorithm			Iterative non-linear regression		
	E_{CE}	d	k	E_{CE}	d	k
Eq. (2)						
3.7	132.0	−16.77	1.16 to 1.27	130.6 ± 0.3	−21.67 ± 2.40	2.08 ± 0.29
3.2	75.0	−37.18 to −51.48	0.50 to 0.76	79.3 ± 0.4	−52.9 ± 0.70	0.59 ± 0.02
2.7	91.4	2.1	0	93.5 ± 0.1	9.01 ± 9.15	6.85 ± 3.92

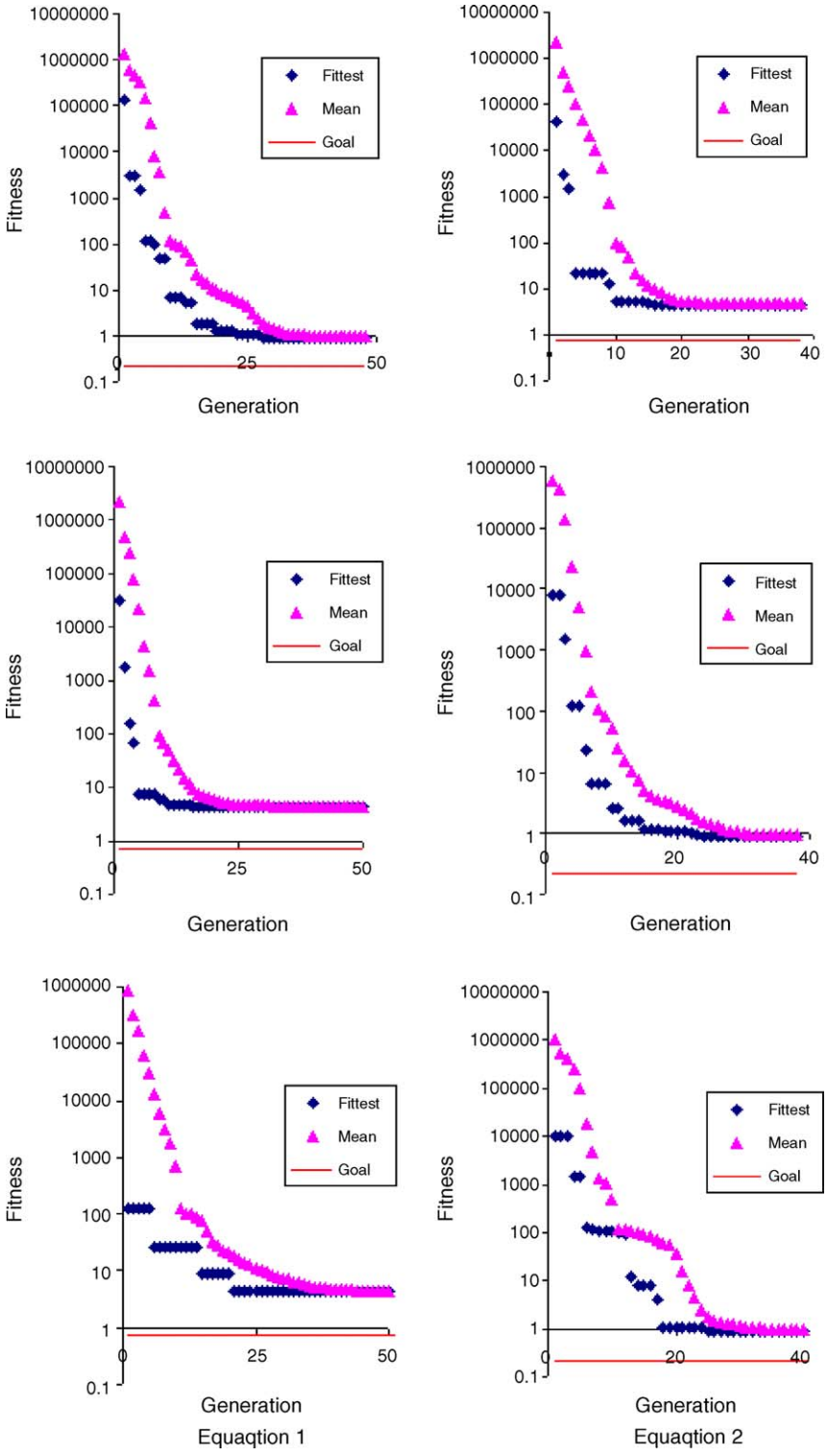

Fig. 4. Genetic algorithm performance plots for the 10^{-3} M calcium ISE response for three independent runs of the genetic algorithm for Eqs. (1) and (2).

local optimizer. The GA though provided initial parameter estimates, which quickly converged to the global optimum. Considerable time savings can be achieved with use of the GA compared to a user trying to find new initial estimates that result in convergence.

4. Conclusion

We have applied a GA to estimate parameters of empirical models for the response curve of a lead, calcium and chloride ISE. The algorithm gave results which were quite close to (if

not within an order of magnitude of) non-linear least square calculated estimates. The GA results were used as initial estimates for non-linear least square regression that converged quite quickly to the global optimum. This approach would be quite applicable to the responses for other types of ISEs as no assumption is made on the response of the ion-selective electrode. Additionally, a GA can be applied to other applications where non-linear regression is needed. GAs can yield estimates, which are close to the global optimum and so allow quick convergence to the final solution set.

References

[1] R.W. Cattrall, W.L. Chek, I.C. Hamilton, P.J. Watkins, Aust. J. Chem. 42 (1989) 1125.
[2] X.D. Wang, W. Shen, R.W. Cattrall, G.L. Nyberg, J. Liesegang, Electroanalysis 7 (1995) 221.
[3] P.J. Watkins, Electroanalysis 9 (1997) 85.
[4] P.J. Watkins, Electroanalysis 13 (2001) 1215.
[5] R. DeMarco, B. Pejcic, X.D. Wang, Lab. Robot. Automat. 11 (1999) 284.
[6] R.S. Ericksen, D. Mackey, P. Alexander, R. DeMarco, X.D. Wang, J. Environ. Mon. 1 (1999) 483.
[7] J. Buffle, N. Parathasarathy, Anal. Chim. Acta 93 (1977) 111.
[8] N. Parathasarathy, J. Buffle, W. Haerdi, Anal. Chim. Acta 93 (1977) 121.
[9] R.J. Leatherbarrow, Trends Biochem. Sci. 15 (1990) 455.
[10] P.J. Watkins, Electroanalysis 17 (2005) 182.
[11] D.B. Hibbert, Chem. Intell. Lab. Syst. 19 (1993) 319.
[12] M. Yang, X.X. Zhang, X. Li, X. Wu, J. Electroanal. Chem. 519 (2002) 1.
[13] T.J. Vandernoot, I. Abrahams, J. Electroanal. Chem. 448 (1998) 17.
[14] I. Rechenberg, Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution., Frommann-Holzboog Verlag, Stuttgart, 1973.
[15] J.H. Holland, Adaption in Natural and Artificial Systems., University of Michigan Press, Ann Arbor, MI, 1975.
[16] M. Maeder, Y.M. Neuhold, G. Puxty, Chemom. Intell. Lab. Syst. 70 (2004) 93.
[17] R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, 2004.
[18] R. Wehrens, H. Putter, L.M.C. Buydens, Chemom. Intell. Lab. Syst. 54 (2000) 35.
[19] W.N. Venables, B.D. Ripley, Modern Applied Statistics with S, 4th ed., Springer-Verlag, New York, 2002.