Development of an empirical model of a nickeliferous chromite leaching system by means of genetic programming

by D.J. Greeff and C. Aldrich*

Synopsis

By making use of genetic programming, empirical models for metallurgical processes can be evolved that are more cost-effective than models determined by means of classical statistical techniques. These methods explore populations of candidate models assembled from sets of variables, parameters and simple mathematical operators, and do not require explicit specification of model structures. The application of the proposed strategies is illustrated by means of a case study pertaining to the leaching of nickeliferous chromite ores. Modelling of the nickel and cobalt extraction from the ores yielded models of similar accuracy compared to those obtained by non-linear regression and artificial neural networks. However, in the case of the iron extraction from the ores, the genetic programming model was significantly more accurate than both the regression and neutral network models.

Introduction

In most processing plants operating conditions have to be adjusted periodically in order to maximize profits or to minimize costs (Seborg *et al.*, 1989). For example, instrumentation needs to be recalibrated and plant units need to be adjusted to accommodate changes in ore feed in mineral processing circuits, blending operations in the petrochemical industry may have to be modified in response to changes in crude oil feedstocks, etc. All these modifications require some form of representation or modelling of the processes, without which adjustment could result in significant inefficiency in overall operations. Moreover, process models can be analyzed to improve understanding of the relevant physical phenomena, while training of process operators to control complex systems or emergency situations can also be facilitated considerably.

Unfortunately, many process systems are not understood sufficiently well to be modelled from first principles. As a result, most of these systems are represented by empirical models based on experimental data. These models typically consist of simple polynomials which relate a set of explanatory variables or process inputs to a set of process responses or output variables. Most empirical models are based on regression analysis to obtain the values of the parameters of the model, $y = f(\mathbf{x},\mathbf{p})$ which is based on the experimental data. Regardless of whether the model is linear or nonlinear, the criterion for estimation of the best model parameters is usually based on minimization of a least squares objective function

$$S = \sum_i^N \left(y_i \pm \acute{y}_i\right)^2 \qquad [1]$$

where y_i is the i'th of N experimental or observed values of the response variable, and y_i is the i'th value predicted by the model. Nonlinear methods are typically based on iterative procedures for estimation of the parameters, such as Gauss-Newton, Marquardt or Powell algorithms (Kuester and Mize, 1973), while in linear methods the parameters can usually be computed direct.

Since the structure of the model is not known beforehand, considerable effort may be required to find a suitable model to which parameters can be fitted. In addition, several assumptions have to be made with regard to the variables and the model, e.g. the distribution of the data (usually assumed to be normal), the independence of the variables, etc. Real world problems seldom meet all these criteria (Howard and D'Angelo, 1995).

Although the need for a priori specification of an explicit model, or other assumptions can be obviated with nonparametric techniques such as artificial neural networks (Haykin, 1994), or multi-adaptive splines (Friedman, 1991), these approaches give little insight into the nature of the problem, since they do not yield explicit models that can be readily analyzed.

* *Department of Chemical Engineering, University of Stellenbosch, Private Bag X1, Matieland, 7602, Stellenbosch, South Africa, Fax +27(21)8082059, E-mail: CA1@maties.sun.ac.za*

 SA ISSN 0038–223X/3.00 + 0.00. Paper received May 1998; revised paper received Jun. 1998.

Development of an empirical model of a nickeliferous chromite leaching system

In this paper it is shown that by making use of genetic programming, empirical models with an interpretable form can be constructed without having to specify a model structure explicitly. These techniques do not only afford a cost advantage over existing model building strategies, but they are also capable of development of optimal models with minimal user interference.

Genetic programming

Genetic or evolutionay programming techniques enable versatile exploration of complex search spaces, based on the Darwinian theory of natural selection (Goldberg, 1989; Davis, 1991). It differs from conventional search strategies which attempt to improve a single solution to a problem, in that they operate on a population of individual solutions, each of which is rated in terms of some relevant optimization criterion. These solutions are typically coded in some form, such as a binary string (chromosome). For example, a function $y = f(x_1,x_2,x_3)$ can be represented in the fixed format y = VARIABLE-OPERATOR-VARIABLE-OPERATOR-VARIABLE, with VARIABLE capable of randomly assuming any of the three variables x_1, x_2 or x_3, and operator capable of assuming any of the values +, -, * or /. This particular format would initially give rise to randomly-generated fixed length strings or chromosomes, each with five variables or genes in fixed positions on the chromosome, while the actual values of these genes would be referred to as alleles.

Examples of these would be $y_1 = x_1{}^*x_1{}^*x_1$, $y_2 = x_1{-}x_2{+}x_3$ or $y_3 = x_1{+}x_1/x_3$. The adequacy of each model can subsequently be evaluated, based on a set of experimental data to which the functions are fitted. For example, if the data x_1, x_2, x_3, and y = 1, 2, 1, and 3 respectively is used as a test case, the squared error $(y_{actual} - y_{model})^2$ generated by y_1, y_2 and y_3 would respectively be 4, 9 and 1. Various measures of the fitness of each individual in the population of three individuals could be used. If the inverse of the error is used, the respective fitness values of each individual, y_1, y_2 and y_3, would be 0.25, 0.11 and 1. In the case of proportional fitness, which is often used in these algorithms, the respective fitness values of y_1, y_2 and y_3 would be 0.25/(0.25+0.11+1), 0.11/(0.25+0.11+1) and 1/(0.25+0.11+1).

These fitness values are subsequently used as a basis for selection and recombination of individuals which are transmitted to succeeding generations. Through parallel operation, several areas of the search space can be explored simultaneously. Entrapment in local minima is avoided by means of genetic operators, such as mutation, while promising regions can be explored more intensely through flexible allocation of the individuals of populations to different regions in the search space.

Three variants of genetic programming are currently studied, viz. genetic algorithms, evolutionary strategies and evolutionary programming (Angeline, 1995). In the best known variant, genetic algorithms, the evolution of models occurs at the level of gene propagation. In evolutionary strategies, individuals are optimised to better exploit their environment, while in evolutionary programming a model of evolution operates on population of individual species competing for shared resources.

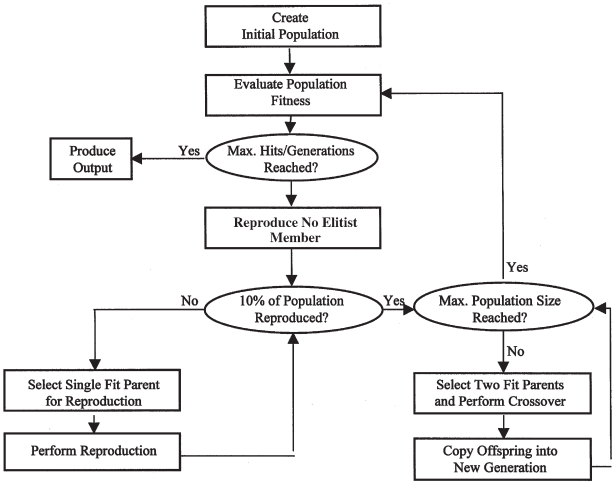

Figure 1—Diagram of evolutionary programming algorithm

In this investigation genetic programming (Koza, 1992) was used to evolve empirical models from experimental data. Although these algorithms typically work on a fixed-length, linear representation of genetic material and operate on this material with fitness proportionate selection, crossover and mutation, Koza (1992) extended this approach to work on hierarchical, tree-structured genetic material, using analogies of the genetic operators such as reproduction, crossover and mutation to perform operations on these structures. The tree structures were encoded in the form of computer subroutines which could be executed on sets of experimental data. This allowed genetic programming to solve problems of unusual complexity.

The particular algorithm that was developed for this study, was based on four steps, namely:

a) Creation of an initial random population of models (encoded in the form of tree-like structures);
b) Calculation of the fitness of each individual (program) within the population;
c) Creation of a new population through the primary genetic operators, crossover and reproduction, based on a fitness selection; and
d) Repetition of steps b) to d) until a satisfactory solution was found.

Note that the mutation operator is not used in this case, since the individuals in the population do not have fixed structures, and the population is therefore not prone to convergence (i.e. all individuals representing identical solutions), as is the case with the use of genetic algorithms. The algorithm is shown in Figure 1 and discussed in more detail below.

Encoding of model solutions

The use of genetic programming to evolve empirical models can best be described by means of a theoretical example. Suppose the relationship between some variable y as an unknown function f(**x**) of a vector of independent or explanatory variables (**x**) has to be determined over an arbitrary region R, based on N exemplars of the form [**x** | y]. Candidate models can consequently be defined in terms of

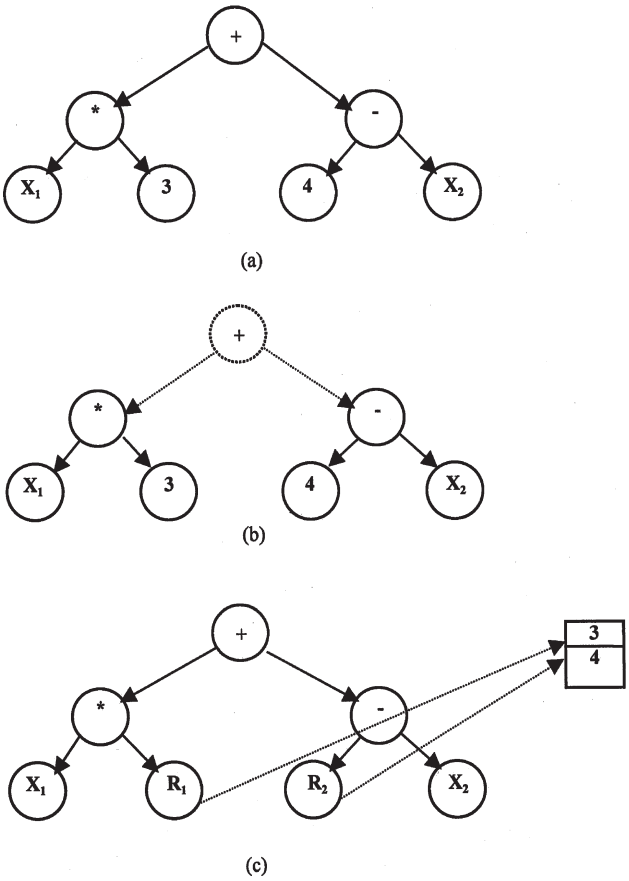

Figure 2—(a) Tree-structure of model equation Y = ((X$_1$*3)+(4-X$_2$)) = 3X$_1$ – X$_2$ +4, and (b) clustered form of tree-structure for model Y = 3X$_1$ – X$_2$ + 4, which means that the model Y = 3X$_1$ – X$_2$ + 4 is essentially treated as two submodels Y$_1$ = 3X$_1$ and Y$_2$ = 4 – X$_2$, which are independently modified by crossover and mutation, after which they are recombined by the operator (broken lines) to reconstruct the final model. In (c) the treatment of model parameters external to tree-structures is shown schematically. The parameters are treated as dummy variables in the evolution of tree-like structures, and are estimated (typically by means of regression) prior to assessing the fitness (accuracy) of the model

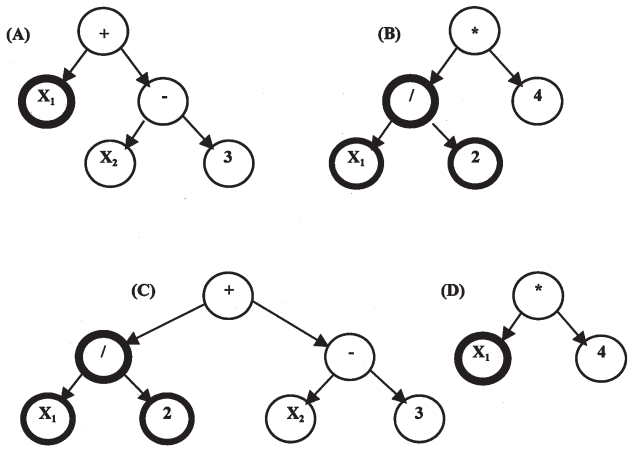

Figure 3—Parents (A and B) selected for the crossover operation with crossover nodes indicated by bold lines. When the two parents exchange genetic material, the X$_1$ terminal node or leaf of parent A (Y$_A$=X$_1$ + X$_2$–3) is exchanged with the (X$_1$/2) subtree of parent B (Y$_B$ = 4X$_1$/2), to yield the offspring C (Y$_C$=X$_1$/2 = X$_2$–3) and D (Y$_D$ = 4X$_1$)

terminal sets T = {**x**, **p**} and functions F = {ψ}, where $\mathbf{x} \in \Re$ represents the set of real explanatory variables in the region R, $\mathbf{p} \in \Re$ a set of real model parameters in the region P, and $\psi \in \wp$ a set of permissible operators. The set of parameter values in the terminal is typically generated randomly during initialization of the programming procedure. This subset of parameter values can consist of be real or integer numbers, and once generated, the set remains fixed. For example, **x** = {x_1, x_2}, **p** = {1} and ψ = {+,-,*,/} imply that models can be generated by addition, subtraction, multiplication and division of the two independent variables x_1 and x_2, and the constant 1. The terminal set can be viewed as the genotype of the population, while the individual members of the population (specific models) can be seen as the phenotypic expression of this genotype.

Tree-structures

The individuals of the population were represented by means of tree-structured codes, such as the one shown in Figure 2. To avoid the generation of unduly complex models, tree-structures were constrained to q levels. A further important modification entailed the use of clusters of trees, rather than single trees themselves. These groups or clusters constituted the individual members of the population and consisted of a number of M trees which represented the model, as shown in Figure 3. During crossover, only a single member of a particular cluster was affected at a time. This had the advantage that alterations to individual members as a result of crossover were restricted, which amounted to a restriction in the amount of variation that a member of a population could undergo at any one time. This is especially beneficial when the best members of the population are close to optimal, since smaller changes to these members allow them to explore their immediate environment in the search space with a higher resolution. Moreover, in certain cases this also facilitated the interpretation of the models, since the final models consisted of the sum of M relatively less complicated terms, each of which could be analyzed individually. After some preliminary runs, the cluster size was fixed at M = 7 and the tree levels at q = 7. These values seemed to give reasonable results, without constraining the trees significantly and were consequently used in all further experimentation, unless otherwise indicated.

Model parameters

Perhaps the most important issue as far as the evolution of models is concerned, is the estimation of the parameter values in the models. Although any set of parameters can, in principle, be generated via the basic set of operators, this is often a very slow process that can have a profound effect on the efficiency of the algorithm. In order to alleviate this restraint on convergence, the parameters can be treated separately from the tree-structures, as indicated in Figure 2c. After crossover, the parameters of descendent trees can consequently be optimised locally by various means. For example, Sharman *et al.*, (1995) have made use of simulated annealing to accelerate the identification of optimal model parameters, while separate optimization procedures based on the use of genetic algorithms have also been proposed (Howard and D'Angelo, 1995). These procedures can accelerate convergence of the algorithms significantly, as

long as they are not too computationally intensive themselves. Although no use has been made of these refinements in this particular investigation, other studies where the model parameters of the descendants of crossover operations are perturbed, have confirmed their advantage.

Generation of initial population and evaluation of fitness

The initial population in each run was created from the terminal and function sets, both with equal probability. The fitness of individuals was evaluated in terms of the number of hits produced by the model, where a hit resulted when the value predicted by the model was within a specified range of the response variable, i.e. if $|y_{pred} - y_{target}| < r$, otherwise it was not a hit. The search terminated when model predictions equalled or exceeded a predefined percentage of hits, or when a previously specified number of generations was reached.

After evaluation of the fitness of the population, the run was initiated by use of the primary genetic operators of reproduction and crossover to create a new population.

Reproduction and crossover

The genetic operators used to evolve new generations were selection, reproduction and crossover. Tournament selection was used, that is the random selection of a group of K individuals from the current population, which had to compete for parenthood (reproduction) on the basis of their fitness values. Only one winner was allowed per group. The values of the genetic operators, as well as the population size were typically determined by a number of initial runs to ensure convergence of the algorithms.

During crossover, which followed reproduction, two parents were selected, with the probability of the selection of any individual proportionate to its fitness. The two parents were then allowed to exchange information, as shown in Figure 3. This exchange was comprised of random selection of cut points on each of the trees, and exchange of the entire subtrees, subject to the constraint that the size of the resulting descendants did not exceed the specified limit (tree depth of q = 7). The parents, as well as their offspring were consequently transmitted to the new generation, which consisted of individuals of which approximately 10% were produced by reproduction alone, while the remaining 90% of the population was created by crossover. This is in contrast with the probabilistic reproduction strategies often used in genetic programming. Moreover, a strategy of elitism was followed, in that the best member of each generation was automatically retained.

These techniques are elucidated by means of the three examples that follow. In the first example simple, known functional relationships are evaluated, after which the technique is compared with regression models of experimental hydrometallurgical systems.

Acid pressure leaching of nickeliferous chromites

Das *et al.*, (1995) investigated the leaching of beneficiated lateritic chromite overburden samples at 250-260°C with 0.3-0.4 g H_2SO_4/g of material in the presence of additives. They have derived quadratic regression equations for nickel, cobalt and iron dissolution (Y_{Ni}, Y_{Co} and Y_{Fe} respectively) at time intervals of 1, 2 and 3 hours as a function of temperature, X_1 (°C), ammonium sulphate concentration X_2 (mol/L) and acid concentration X_3 (mol/L).

The following models were obtained for each time interval for the respective amounts of nickel (Equations 2a-2c), cobalt (Equations 3a-3c) and iron (Equations 4a-4c) leached.

Nickel

$$Y_{Ni(1h)} = 78.5 + 6.75X_1 - 5.5X_2 + 4.0X_3 + 5.0X_2X_3 \quad [2a]$$

$$Y_{Ni(2h)} = 85 + 3.6X_1 - 6.0X_2 + 7.75X_3 + 6.25X_2X_3 \quad [2b]$$

$$Y_{Ni(3h)} = 89.75 - 6.0X_2 + 6.5X_3 + 6.25X_2X_3 \quad [2c]$$

Cobalt

$$Y_{Co(1h)} = 80 - 5.6X_2 + 3.025X_3 + 5.85X_2X_3 \quad [3a]$$

$$Y_{Co(2h)} = 81.8 - 7.7X_2 + 4.25X_3 + 4.9X_2X_3 \quad [3b]$$

$$Y_{Co(3h)} = 85.16 - 8.73X_2 + 5.44X_3 + 3.5X_2X_3 \quad [3c]$$

Iron

$$Y_{Fe(1h)} = 4.42 + 0.67X_1 - 0.78X_2 + 0.76X_3 + 0.69X_2X_3 + 0.09X_1X_3 \quad [4a]$$

$$Y_{Fe(2h)} = 5.20 + 0.72X_1 - 1.255X_2 + 1.08X_3 + 0.68X_2X_3 + 1.1X_1X_3 \quad [4b]$$

$$Y_{Fe(3h)} = 5.61 + 0.74X_1 - 1.635X_2 + 0.99X_3 \; 0.46X_1X_2 + 1.0675X_1X_3 + 0.6425X_2X_3 \quad [4c]$$

As before, runs were conducted with populations consisting of 2000 individuals each. The terminal and function sets were defined as T = {X_1, X_2, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9} and F = {+, -, *, /}, while the number of runs was set at 100 generations. In all cases tournament selection involving 7 parents was used, with a 10%-90% reproduction to crossover ratio for population make-up, as described earlier. For both the cobalt and the nickel data sets, the hit range was defined as less than 1 percentage point from the target value. For the experimental data pertaining to the leaching of the iron, the range was defined as 0.5 percentage points, since the experimental data had a significantly smaller range in this case.

Since the data used to evolve models were sparse, the models were evaluated by means of cross-validation. The residuals generated in this way give an indication of the ability of the model to generalise the underlying trends reflected by the data, and were generated as follows. One test data point was removed from the data set, and the data set was subsequently used to evolve the model (typically by a least squares procedure). The fitted model was then tested against the point originally omitted from the data set and the residual calculated. The test point was subsequently returned to the data set, while a new observation not used before for testing purposes was selected. The model was again fitted to the remainder of the data, and tested against the test data point not used to estimate the model parameters. This procedure was repeated until all the data points had been tested. This ensured that data were not used simultaneously for evolution of the model structure and fitting of model parameters, as well as for validation of the model.

Tree-structures consisting of seven clusters each were used, and the model represented by Equation 5 is a typical result,

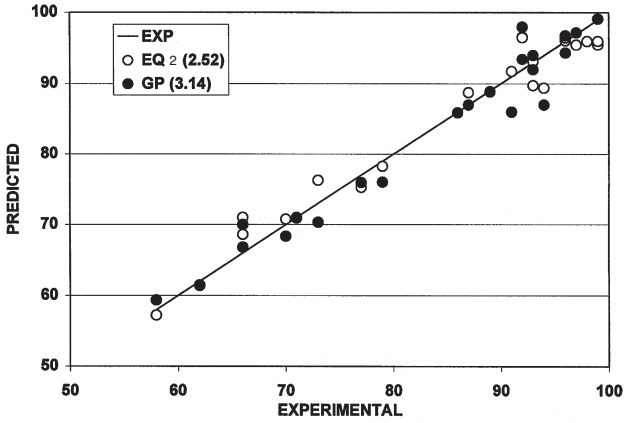

Figure 4—Extraction of nickel. Solid line represents experimental data, empty circles represent separate models (Equations 2a–2c) obtained by Das *et al.*, (1995), and solid circles represent separate models obtained by evolutionary programming. RMS errors for each model are indicated in brackets in the legend

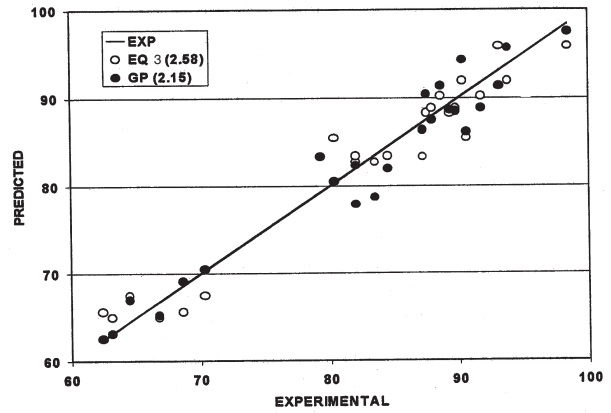

Figure 5—Extraction of cobalt. Solid line represents experimental data, empty circles represent separate models (Equations 3a–3c) obtained by Das *et al.*, (1995), and solid circles represent separate models obtained by evolutionary programming. RMS errors for each model are indicated in brackets in the legend

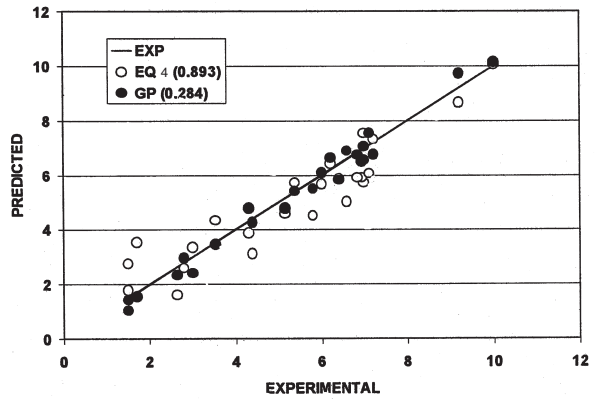

Figure 6—Extraction of iron. Solid line represents experimental data, empty circles represent separate models (Equations 4a–4c) obtained by Das *et al.*, (1995), and solid circles represent separate models obtained by evolutionary programming. RMS errors for each model are indicated in brackets in the legend

$$Y_{Co(1h)} = 8 + (((9-X_2) + (X_3/7))*7) + (((X_1+4)-2) + (X_2+((X_1*X_2) + (X_1+X_3)))) + X_2*X_3 + (((X_3*X_3)+(X_2*X_3))*(X_3+(((X_2/X_3)-X_1)+4))) \quad [5a]$$

This is equivalent to

$$Y_{Co(1h)} = 73+2X_1-8X_2+2X_3+X_1X_2+6X_2X_3+X_2^2+4X_3-X_1X_2X_3-X_1X_3^2+X_2X_3^2+X_3^3 \quad [5b]$$

On the whole the models generated by genetic programming showed a significant correspondence with the models proposed by Das *et al.*, (1995). For example, the percentage of the cobalt leached after one hour as predicted by Das *et al.*, (1995) (Equation 3a), corresponds closely with the evolved model (Equation 5b). Although the evolved model is a third-order polynomial, the constants differ by 10% only, while the coefficients of X_2 are -5.6 and -8 for the model of Das *et al.*, (1995) and genetic programming respectively. Similarly, for X_2 it is 3.025 and 2 respectively, while for the interaction term it is 5.85 and 6 respectively. The models for the other equations can likewise be compared.

In addition, a second series of runs based on terminal and function sets including time as a variable T = {t, X_1, X_2, X_3, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9} and F = {+, -, *, /} was conducted. Figures 4, 5 and 6 show the target values (solid line), the values predicted by the regression models of Das *et al.*, (1995) (the empty circles), as well as the values predicted by the evolved models (the solid circles) for the nickel, cobalt and iron leaching systems respectively, where the terminal sets are expanded to incorporate time. The values in brackets in these figures are the root mean square (RMS) values of the prediction errors in the case of the models of Das *et al.*, (1995), and cross-validation residuals in the case of the models derived by means of the evolutionary programming algorithm.

If anything, a comparison of these results discriminate against the evolutionary programming models, since somewhat less information was used to construct the genetic programming models, than was the case with the regression models, where parameter estimation was based on all the experimental data. As shown in Figures 4-6, the evolved models (only one per system) had an accuracy similar to the regression models (three separate models per system) in the case of leaching of the nickel and cobalt. In contrast, the models evolved were significantly more accurate than the combined regression models of Das *et al.*, (1995) in the case of leaching of the iron.

Owing to the simplicity of the terminal set, solutions consisting of rather complex equations were generated. These equations were typically of the form $Y_{(j=Ni,Co,Fe)} = \Sigma_{i=1}^{M} \alpha_i^j(\mathbf{X})/\beta_i^j(\mathbf{X})$, where α_i^j and β_i^j represented multivariate polynomials of orders up to 4 in the explanatory variables **X**, which are difficult to interpret and can be seen as equivalent to non-parametric models. This can be shown by way of example in Equation 5c, for cobalt, where $Y_{Co} = f(X_1, X_2, X_3 X_4)$. As with other non-parametric or weakly parametric models, it is important that overfitted models are checked, since these types of models tend to have a large number of parameters.

$$Y_{Co} = 2X_4/(X_2X_3-X_1-5) + X_2X_3 + X_2^2/(4X_2X_4^2+X_1X_4^2-X_2X_4^2) + 5X_3^2-X_+X_3^2 + X_2^3/(4X_2X_4^4 + X_1X_4^4 - X_2^2X_4^4) + X_3/X_1 + X_3/(X_1+X_3) - X_3^2 - X_1X_3 \quad [5c]$$

Figure 7 gives an example of the convergence of the

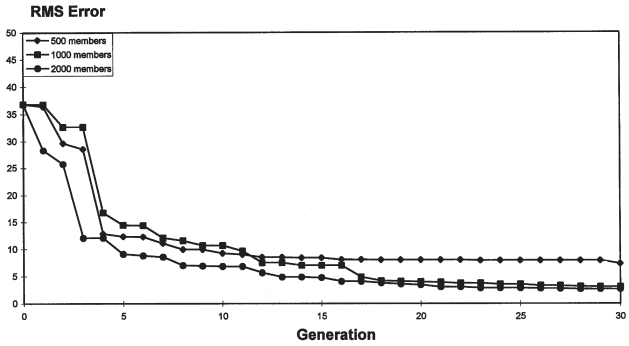

Figure 7—Convergence of evolutionary programming technique during derivation of nickel leaching model

evolutionary programming technique for the derivation of the nickel leaching models with different population sizes. As can be seen from Figure 7, there was little difference in the RMS error values of the models derived with populations of 1000 or 2000 members after approximately 30 generations. Similar profiles were obtained for the cobalt and iron.

Discussion of results

One of the main difficulties of model development with genetic programming is the proliferation of complex tree-like structures which are difficult to interpret, and which tend to compromise the evolution of generalised models (Nordin and Banzhaf, 1995). Several approaches can be followed to curtail tree size explosion. For example, a penalty factor can be incorporated into the fitness function, which is proportional to the size of the tree. Preliminary work using fitness functions divided by the apparent size of a tree, that is the number of operators or parameters in the equations prior to simplification, appeared to alleviate the problem. However, this approach is not reliable, since the *apparent* tree size may not be a good indication of the inherent complexity of the model (the sizes of simplified trees are better indications, but simplification of the trees can be difficult).

In the example considered in this paper, the regression models could predict the dissolution of the nickel, cobalt and iron with root mean square (RMS) errors of respectively 2.52, 2.58 and 0.893, as indicated in Table I. The genetic programming models correspondingly predicted the percentage dissolution of the nickel, cobalt and iron with root mean square errors of 3.14, 2.15 and 0.284. For comparative purposes, a sigmoidal backpropagation neural network (Haykin, 1994) with 4 input nodes, 2 hidden nodes and one output node were also trained on the data. Since few exemplars were available, training was based on 5-fold cross-validation, i.e. the net was trained on 19 examplars at a time, and validated against the remaining five. The result RMS errors are included in table 1. An analysis of variance has shown that the differences in the squared errors in the case of nickel (F-ratio = 1.292, significance level = 0.262) and cobalt (F-ratio = 0.127, significance level = 0.723) were not significant (i.e. a significance level exceeding 0.05). In contrast, the squared errors generated by the genetic programming model in the case of iron (F-ratio = 13.749, significance level = 0.001) were highly significant (i.e. a significance level of less than 0.01).

Table I

Summary of results obtained with genetic programming (GP), artificial neural network (ANN) and regression models

Model	Data	RMS-values	Sig. Level (F-ratio): GP/eq.	Sig. Level (F-ratio): GP/ ANN
GP/ANN/ eq. 1	(Ni)	3.14/2.55/ 2.52	0.262 (1.292)	0.273 (1.230)
GP/ANN eq. 2	(Co)	2.15/2.93/ 2.58	0.723 (0.127)	0.356 (0.868)
GP/ANN eq. 3	(Fe)	0.284/0.970/ 0.893	0.001 (13.75)	0.001 (12.86)

Likewise, an analysis of variance has shown that the differences in the squared errors of the GP models and the neutral network models in the case of nickel (F-ratio = 1.230, significance level = 0.273) and cobalt (F-ratio = 0.868, significance level = 0.356) were not significant (i.e. a significance level exceeding 0.05). In contrast, the squared errors generated by the gentic programming model in the case of iron (F-ratio = 12.86, significance level = 0.001) were highly significance (i.e. significance level of less than 0.01).The main difficulty that had to be overcome with regard to the algorithm implemented as described above, concerned the evolution of numerical values for the model parameters. These values were evolved through subtrees, which were in principle able to produce any rational number desired. However, in practice the trees tended to grow disproportionately large (and their structures difficult to interpret), while runs were more likely to fail to converge. In addition, the implicit evolution of appropriate parameters by means of the evolutionary algorithm used in this investigation (Figure 1) was computationally intensive.

Much can therefore be gained by separating the parameter search from the variable search or model structure evolution, and to determine the parameters by means of standard methods, such as gradient descent techniques, simulated annealing (Sharman *et al.*, 1995) or genetic algorithms (Howard and D'Angelo, 1995).

Conclusions

In this paper empirical models were developed with evolutionary algorithms and the following conclusions can be made.

- The models evolved were either of comparable accuracy or significantly more accurate than those which were developed by means of standard least square methods artificial neutral networks
- Most of the search time spent on the development of the models was consumed by the evolution of appropriate parameter values for the models, while useful model structures could be identified relatively rapidly.
- Since the evolved models were derived from terminal sets containing only the most basic operators, their structures could be simplified to ratios of polynomial functions. The interpretation of such functions can be

complicated and in this sense the evolved models were equivalent to other non-parametric models, such as neural networks or kernel-based regression models.

➤ Convergence could be accelerated considerably when the initial population was seeded with more structured terminal set elements (such as linear models of the system, even if these did not provide a good fit).

Nomenclature

F	functional set of operators from which models are constructed
K	number of individuals involved in tournament selection
M	number of clusters in a tree
N	total number of exemplars available for construction of models
P	range of values assumed by parameters, **p**
p	initial set of parameters available for model evolution
q	depth of a tree-like structure, i.e. a measure of the complexity of evolved models
R	range of values assumed by explanatory variables, **X**
S	sum of squared residuals of a model, i.e. $S = \Sigma_i^N (y_i - \hat{y}_i)^2$
T	terminal set of variables and parameters from which models are constructed
t	time
x, X	explanatory or independent variables in general, as defined in the text
y, Y	response or dependent variable in general, as defined in the text
$Y_{Co(jh)}$	percentage of cobalt leached after j hours
$Y_{Fe(jh)}$	percentage of iron leached after j hours
$Y_{Ni(jh)}$	percentage of nickel leached after j hours
y_i	i'th observation of a response
$\hat{y}_i$	i'th prediction or estimate of a response
α_i^j	i'th part in the numerator of a composite polynomial function for j'th metal species in an evolved leaching model
β_i^j	i'th part in the denominator of a composite polynomial function for j'th metal species in an evolved leaching model
ψ	set of mathematical operators on which evolution of models is based

References

Angeline, P.J. Evolution revolution: An introduction to the special track on genetic and evolutionary programming. *IEEE Expert*, 10(3), 1995. pp. 6–10.

Das, G.K., Acharya, S., Anand, S. and Das, R.P. Acid pressure leaching of nickel-containing chromite overburden in the presence of additives. *Hydrometallurgy*, 39, 1995. pp. 117–128.

Davis, L. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.1991.

Friedman, J.H. Multivariate Adaptive regressions splines. *Annals of Statistics, 19*. 1991. pp. 1–141.

Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts. 1989.

Haykin, S. *Neural Networks*. MacMillan College Publishing Company, Inc., Englewood Cliffs, NJ, USA, 1994.

Howard, L.M. and D'Angelo, D.J. The GA-P: A genetic programming hybrid. *IEEE Expert*, 10(3), 1995. pp. 11–15.

Koza, John R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge, Massachusetts. 1992.

Kuester, J.L. and Mize, J.H. *Optimization techniques with Fortran*. McGraw-Hill Book Co., N.Y. 1973.

Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*, 2nd ed., Springer-Verlag, New York.1994.

Nordin, P. and Banzhaf, W. Explicitly defined introns and destructive crossover in genetic programming. *Proceedings of the 6th International Conference on Genetic Algorithms* (ICGA95), San Francisco, CA, Morgan Kaufmann Publishers. 1995. pp. 310–317.

Seborg, D.E., Edgar, T.F. and Mellichamp, D.A. *Process Dynamics and Control*. John Wiley and Sons, Inc., New York. 1989.

Sharman, K.C., Esparcia Alcázar, A.I. and Li, Y. Evolving signal processing algorithms by genetic programming. *Proceedings of IEE/IEEE Genetic Algorithms in Engineering Systems: Innovations and Applications* (GALESIA). 1995. ◆

Wits school gains international recognition*

A reputation in powder metallurgy that is growing in stature all over the world has led to Professor Silvana Luyckx of the School of Process & Materials Engineering at the University of the Witwatersrand receiving a grant from Germany and an invitation to the United States.

Professor Luyckx specializes in hardmetals and has spent several years developing uses for vanadium carbide. As the largest vanadium producer in the world, this work is of economic importance to South Africa.

Local vanadium producers have sponsored the advanced research being done at the school. This has resulted in increased awareness overseas of the material and its potential in the market.

Grant

Professor Luyckx has received a three-year grant from Germany which provides the school with two full bursaries for South African students to carry out postgraduate work relevant to the local hardmetal industry.

The grant was secured by Professor Dr Ing Hans Sockel of the Institute for Material Research at the University of Erlangen-Nürnberg. Professor Sockel submitted a German-South African science and technology research proposal titled *Improvement of Hardmetals* to his government which approved the pilot project.

Professor Luyckx's role will be in the planning, research and interpretation of the results and she will monitor the students' progress. Two students have been selected by the school and final arrangements are in progress.

United States

An invitation from the Office of Naval Research Europe of the Department of the United States Navy has been extended to Professor Luyckx to visit several American universities in May as well as attend $PM^2TEC'98$, the 1998 International Conference on Power Metallurgy & Particulate Materials. The conference, organised by the Metal Powder Industries Federation, will be held in Las Vegas from 31 May to 4 June.

During her trip, Professor Luyckx will visit Rutgers University in New Jersey, Penn State University in Pennsylvania and the University of Boston in Maryland. Among others, she will work closely with Professor Bernard Kear of Rutgers University on finding other applications for vanadium.

It seems likely that Professor Luyckx will incorporate a visit to Venezuela in her return trip. Intevep SA, a group giving technological support to research associated with Venezuela's oil industry, has indicated a strong interest in consulting her about uses for the vanadium in Venezuelan oil. ◆

* *Issued by: Lynne Hancock Communications, P.O. Box 180, Witkoppen, 2068, Tel: (011) 460-1000. E-mail: Ihcom@icon.co.za*

Richards Bay Minerals (Tisand) awarded ISO 14001 accreditation*

Richards Bay Mineral's Tisand operation, a world leader in the recovery and processing of rutile and zircon sands, and ilmenite for further benefication, has been awarded the internationally recognized ISO 14001 accreditation applicable to environmental management.

At the presentation of the certificate by The South African Bureau of Standards (SABS) on 7 July 1998, RBM managing director Keith Rumble said that this achievement, the second in the South African mining industry, reflects the extent to which RBM's shareholders and employees strive for excellence to ensure the protection of the environment.

'Environmental management has been at the heart of RBM's mission since inception, and our dune rehabilitation practices have received local and international recognition. Although we have always had a good environmental management programme, we decided to embark on ISO 14001 accreditation to formalize and integrate our approach to environmental, safety and quality management.

Richards Bay Iron and Titanium (RBIT), the other company under the RBM umbrella which is responsible for the benefication of ilmenite to produce titania slag and pig iron, is in the process of obtaining its ISO 14001 accreditation.

The organization already has ISO 9002 accreditation for the whole operation and the coveted five-star rating from NOSA (National Occupational Health and Safety Association) for occupational health and safety management,' said Keith Rumble. ◆

* *Issued by: Piet Maré, Head of Public Relations, Richards Bay Minerals. Tel: (0351) 901 3440 Fax: (0351) 901 3442.*