Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Information Sciences 150 (2003) 95–117

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Symbolic and numerical regression: experiments and applications

J.W. Davidson, D.A. Savic *, G.A. Walters

Department of Engineering, School of Engineering and Computer Science, University of Exeter, Harrison Building, North Park Road, Exeter, Devon, EX4 4QF, UK

Received 5 August 2000; accepted 7 April 2001

Abstract

This paper describes a new method for creating polynomial regression models. The new method is compared with stepwise regression and symbolic regression using three example problems. The first example is a polynomial equation. The two examples that follow are real-world problems, approximating the Colebrook–White equation and rainfall-runoff modelling. The three example problems illustrate the advantages of the new method.

Keywords: Genetic programming; Least squares; Rule-based programming; Stepwise regression; Symbolic regression

1. Introduction

This paper describes a new regression method for creating polynomial models. The new technique, referred to as the hybrid method, combines the parameter optimisation of numerical regression methods with the evolutionary search of symbolic regression. The hybrid method is compared with two established regression methods, symbolic regression and stepwise regression in three example problems.

* Corresponding author. Tel.: +44-1392-263-637; fax: +44-1392-217-965.
E-mail address: d.savic@exeter.ac.uk (D.A. Savic).

0020-0255/02/$ - see front matter
PII: S0020-0255(02)00371-7

1.1. Modifications to symbolic regression

The new regression method includes changes to the basic genetic programming algorithm first proposed by Koza [1]. There are several difficulties with symbolic regression based on genetic programming. The original technique makes use of non-adjustable constants, referred to as ephemeral random constants. The constants do not necessarily assume optimal values as in numerical regression methods. The use of non-adjustable constants contributes to the complexity of expressions as well as the inaccuracy. Previous research has proposed methods to improve on ephemeral random constants. McKay et al. [2] included two adjustable parameters in each expression and Watson and Parmee [3] used micro-evolution to evolve constant values within expressions. The algorithm described in this paper uses the method of least squares to obtain parameter values.

Another problem with symbolic regression is "bloat" that results from the inclusion of non-functional code, or introns, within expressions (see Refs. [4,5]). Code bloat causes the genetic programming search process to become ineffective after approximately 50 generations. (The limit of 50 generations is a crude generalisation. For more information on introns and code bloat, see Ref. [4].) Smith [5] describes categories of introns and explains why bloat occurs. Smith [5] also describes a variety of approaches to controlling code growth including the prevention and removal of introns, parsimony pressure in the fitness function, restricting crossover, and the use of alternative selection schemes. Babovic and Keijzer [6] and Keijzer and Babovic [7] have developed a method that improves on symbolic regression by evaluating the dimensions or units of the resulting output value. The method requires solution equations to approximate the dimensions as well as the values of target data. The method also includes a technique that fine tunes constant values. Although the method is specifically aimed at producing mathematical models that are more logical and coherent, the approach reduces the complexity of resulting expressions and the problems associated with code bloat as a side effect.

In addition to non-adjustable constants and code bloat, the original symbolic regression method can produce expressions with hidden complexity. Due to the property of distribution of multiplication into addition, expressions created by symbolic regression can undergo a combinatorial explosion in the number of terms in the expression when expanded algebraically. The combinatorial explosion can be so severe that the number of parameters in regression equations produced by the algebraic expansion of these expressions can greatly exceed the number of observations in large data sets. A method that uses adjustable parameters in place of ephemeral random constants runs the risk of producing underdetermined functions due to this problem.

2. Methods

2.1. The hybrid method

The new method described in this paper, referred to as the hybrid method, overcomes the problems of non-adjustable parameters, code bloat and the creation of underdetermined functions. However, the new approach restricts the wide range of operators normally used in symbolic regression to a subset consisting only of addition, multiplication and non-negative integer powers. The expressions that result from the limited set of operators are forms of polynomials. A rule-based program consisting of 56 rules algebraically transforms all expressions produced through the evolutionary process to the form of the right hand side of Eq. (1):

$$\hat{y}_i = \sum_{i=1}^{n} a_i z_{ij} \tag{1}$$

where $\hat{y}_i$ is the value returned by the expression for the jth observation with independent variables $\boldsymbol{x}_j = \langle x_{1j} x_{2j} \ldots x_{dj} \rangle$; a_i is an adjustable parameter for the ith term in the expression; z_{ij} is a transformed variable, a unique product of the independent predictor variables $\boldsymbol{x}_j = \langle x_{1j} x_{2j} \ldots x_{dj} \rangle$ raised to non-negative integer powers; n is the number of terms in the expression; and d is the number of independent predictor variables.

In addition to transforming all expressions to the polynomial form of Eq. (1) the rule-based program eliminates all non-functional code (introns) produced by evolutionary operations. Non-functional code includes expressions such as terms formed by the product of zero and other coefficients. The rule base simplifies expressions by evaluating terms and coefficients that consist entirely of constants and replaces them with a single constant where possible. Once the rule base has transformed the expressions to the required polynomial form the program computes the optimal value for constants in the expression (adjustable parameters a_i in Eq. (1)) by the method of least squares. Davidson et al. [8] provides a description of the rule-based program and methods for optimising adjustable constants.

Experience has shown that the methods for generating starting solutions normally used in genetic programming, such as ''ramped half'' (see Ref. [1]), occasionally produce solution equations that have an extremely large number of terms when expanded algebraically due to the combinatorial problem explained previously. To avoid the problem a new method for generating starting solutions creates short, two-term expressions in the form of Eq. (1). With successive generations the equations grow to a maximum length specified by the user in advance. The hybrid regression method also uses modified operations of crossover and mutation rather than the conventional genetic

programming operations. The new crossover operation cannot operate at any location in the parse tree of an expression. Instead the new operation divides expressions between terms in the summation and cannot operate within individual terms. New mutation operations act to modify power values within terms and occasionally create entirely new terms at random. The new evolutionary operations avoid the problem of creating excessively large expressions after algebraic expansion by the rule base. Solutions produced by the new operators exhibit greater similarity to their parent solutions. The new crossover operation produces child solutions that are a sum of terms found in either of the two parents. The new crossover operator can also impose limits on the number of terms that appear in child solutions and thereby restrict the maximum size of expressions, controlling the rate by which the length of expressions grows.

As mentioned previously the expressions in the starting population consist of only two terms. As the search progresses the maximum number of terms gradually increases causing the length of expressions to increase to a specified maximum number of terms. Expressions are assessed on the basis of mean square error (MSE). The program retains the best expression found for each length from the best two term expression up to the best expression with the maximum number of terms specified by the user. Accuracy of the best solutions, as measured by MSE, improves with increasing length. This set of best solutions represents the trade-off between accuracy and complexity (or computational effort).

2.2. Stepwise regression

Stepwise regression is a popular and highly effective method for building regression models. Draper and Smith [9] describe the stepwise regression algorithm in greater detail and regard it as "one of the best variable selection procedures." The stepwise regression algorithm constructs the model through a series of iterations. Each iteration consists of one of two procedures that either adds a term to the model (referred to here as the selection procedure) or removes a term (elimination). After a term is added or removed stepwise regression optimises the values of coefficients (represented as a_i in Eq. (1)) using the method of least squares similar to the hybrid regression method described in the previous section. The user of stepwise regression must specify a list of transformed variables (vectors $\boldsymbol{z}_i$ of which the terms z_{ij} in Eq. (1) are members) that the procedure uses in creating the regression model. The transformed variable added by the selection procedure is the term from the list that produces the model with the lowest MSE. Stepwise regression can be used to construct expressions similar to Eq. (1) if the transformed variables in the list consist of products of integer powers of the independent variables, $\boldsymbol{x}_j$. The use of products of integer powers is not a necessary limitation of stepwise regres-

sion. The transformed variables are restricted to integer powers for the examples presented in the paper to facilitate the comparison by restricting the two methods to the same search space.

There are two approaches to stepwise regression, (1) stepwise selection and (2) forward selection. Both methods begin by creating the best first-order linear regression equation consisting of two terms, a single transformed variable and a constant. As mentioned previously with each iteration the selection procedure adds a term consisting of the product of a constant and a transformed variable ($a_i z_{ij}$) to the model. The transformed variable selected is the one that produces the greatest reduction in MSE.

Forward selection is the simpler form of stepwise regression. Both methods add terms to the model from the list by iterations of the selection procedure. Stepwise selection, the more complex procedure, can also remove terms on the basis of t-statistics (or equivalent partial F-statistics) provided for each term by the method of least squares (described in Ref. [9]). After each selection step the elimination procedure of stepwise selection can remove the least significant term from the regression model if one or more t-statistic falls below a minimum level of significance specified by the user, usually corresponding to 95% or 99% confidence. The algorithm then calculates a new set of coefficients and t-statistics and the elimination procedure repeats until all terms remaining in the model exceed the minimum significance level. The forward selection method also makes use of t-statistics. However, it examines only the most recent term to enter the model. In the case of both methods if the most recently added term falls below the minimum significance level the algorithm must terminate to avoid cycling.

Stepwise selection is the more powerful of the two methods because the significance of terms can change as new terms enter the model. A term that was highly significant in a previous iteration can become superfluous as the model grows. Forward selection is a simple best-first search or hill climbing technique. Methods of this type are notorious for becoming trapped in local optima. Although stepwise selection is able to undo selections made in previous iterations it too can become trapped in a local optimum, as the first example problem demonstrates.

2.3. Symbolic regression

The symbolic regression procedure is described in Ref. [1]. As mentioned previously the method makes use of ephemeral random constants that have values determined on initialisation, which can only be modified through mutation. The method can make use of non-linear equation forms, a wide variety of function types and even programming constructs such as "if-then" statements. The commercial software package known as Discipulus Pro (Ref. [10]) was used to obtain the symbolic regression results for the first two example

problems in this paper. The third example used software developed by Polyhonen and Savic [11]. Discipulus Pro implements the standard symbolic regression algorithm with several enhancements including the ability to produce code directly in machine language. Machine language expressions and the absence of adjustable parameters result in very fast evaluation of expressions. Execution times were a small fraction of those associated with stepwise regression or the hybrid method. Searches conducted with symbolic regression involved much larger populations and many more solution evaluations.

3. Example 1: a polynomial problem

This section of the paper compares the three methods using an example problem for which the solution equation is known in advance. The data set for a sample problem consists of three independent variables x_1, x_2 and x_3 with values selected randomly on an interval between 0 and 1, listed in Table 1. Eq. (2) is the equation used to calculate the values of the dependent variable, y.

$$y = 3x_1^3x_2^2x_3 + 4x_1^2x_2^2x_3 + 2x_1x_2^3x_3^2 \tag{2}$$

Table 1
Data points for example 1

x_1	x_2	x_3	y
0.003198	0.050759	0.713179	5.01E−07
1.057343	0.026417	0.395321	0.002218
1.56941	1.752838	1.964279	194.6688
1.410636	1.121589	0.116711	2.459192
0.953305	0.017604	0.269427	0.000521
0.143245	1.561776	1.727959	3.641707
0.113425	1.345502	0.84109	0.475935
1.832543	1.701547	1.999461	256.8236
0.134863	1.506071	1.524166	2.4175
1.827083	1.728888	1.992118	263.406
1.425313	1.185431	0.265792	6.615073
0.19024	1.805963	1.820947	8.413532
1.955725	0.632706	0.933749	14.97103
0.514412	1.441612	1.255968	8.691103
1.539753	1.651683	1.954597	161.9761
0.873072	0.126997	1.450225	0.125535
0.627877	0.954501	0.016725	0.035649
1.977713	0.337065	1.970665	9.286923
1.736231	1.985806	0.219122	25.29245
1.199153	0.30488	1.997693	2.299873

3.1. Combinatorics of the polynomial problem

The number of transformed variable terms available to stepwise regression and the hybrid regression method depends on a maximum power value specified by the user. For this demonstration a maximum power value of 6 is arbitrarily selected for both stepwise regression and the hybrid method. The number of candidate transformed variables (342) is $(m+1)^d - 1$ where m is the maximum power value and d is the number of independent predictor variables.

For the first example problem it is possible to calculate the probability of generating the correct solution at random. If an expression contains all of the key terms least-squares optimisation reduces the coefficients of any other terms to zero. It should be noted that superfluous coefficients are reduced to zero only if the data do not contain random noise, the target function is polynomial and all the key terms are included in the equation. When n is greater than the length of the solution expression there are many possible expressions that represent optimal solutions. The formula for the number of optimal expressions, N_0, with maximum length n is:

$$N_0 = \sum_{i=n_t}^{n} \binom{(m+1)^d - k - 1}{i - k - 1} \tag{3}$$

where k is the number of key terms; and n_t is the length of the target expression.

The number of unique expression forms N with maximum length n is given by:

$$N = \sum_{i=2}^{n} \binom{(m+1)^d - 1}{i - 1} \tag{4}$$

The probability of generating the optimal equation at random is represented by N_0/N. The probability improves with larger equations.

3.2. Results of the hybrid method

Three different values of maximum length of expressions were used with the hybrid regression method. The values consisted of maximum lengths of 5, 7 and 14 terms. Table 2 lists the number of solutions required to produce the optimal solution with each trial. The hybrid regression method was able to find the correct solution by generating relatively few solutions. The rows at the bottom of Table 2 list the mean, maximum and minimum number of solutions generated for each of the three maximum lengths. The bottom row lists the ratio of non-optimal solutions to optimal solutions, which represents the inverse of the probability of producing the solution by random generation. As the maximum length restriction increases the probability of generating the optimal solution at random improves greatly although the mean number of

Table 2
Number of solutions required by hybrid method for example 1

Trial	Solutions required to find optimum		
	5 term maximum	7 term maximum	14 term maximum
1	1757	7052	2103
2	3502	5374	1617
3	3222	2183	1230
4	3558	1787	2058
5	1847	1782	2184
6	3011	2183	3213
7	2672	931	9879
8	1717	1064	1568
9	6697	1558	1789
10	2117	10,262	2294
Mean	3010	3417.6	2793.5
Maximum	6697	10,262	9879
Minimum	1717	931	1230
N/N_0	1,666,895	333,414	23,325

solutions required to find the optimal solution remains relatively unaffected. For a maximum length of five terms the probability of producing the optimal solution at random is 1 in 1,666,895 and the hybrid method required an average of 3010 evaluations to find the correct form.

3.3. The results of stepwise regression

Table 3 shows the results of forward selection and stepwise selection. Both methods use 95% as the minimum significance for terms. Both methods fail to find the optimal solution to the problem. Stepwise selection terminates at an 11 term model. The next term to enter falls below the 95% minimum significance level. Forward selection includes "significant" terms until it reaches the limits imposed by 20 data points. Forward selection models can include terms that are below the 95% significance level. However, terms that enter must be above 95% significance during the iteration in which they are included.

Fig. 1 shows the value of MSE for the iterations in which the two stepwise regression methods, stepwise selection and forward selection, produce different results. The horizontal axis of Fig. 1 represents the number of parameters (or the number of terms) in the expression and the vertical axis is the MSE. The solid line represents solutions created by stepwise selection and the dashed line represents forward selection. As Table 3 and Fig. 1 show, stepwise selection and forward selection produce the same solutions for the first eight iterations, at which point the solution consists of nine terms. Two terms in the nine term expression have t-statistics that fall below the minimum significance level. At this point the two methods begin to produce different solutions. The forward

Table 3
Error and computational effort for stepwise regression in example 1

Iteration	Forward selection			Stepwise selection		
	Terms	MSE	Solutions	Terms	MSE	Solutions
1	2	9.43321	342	2	9.43321	342
2	3	2.27646	683	3	2.27646	683
3	4	0.526472	1023	4	0.526472	1023
4	5	0.124819	1362	5	0.124819	1362
5	6	0.067219	1700	6	0.067219	1700
6	7	0.026446	2037	7	0.026446	2037
7	8	0.014747	2373	8	0.014747	2373
8	9	0.012181	2708	9	0.012181	2708
9	10	0.008397	3042	8	0.012016	3044
10	11	0.001344	3375	9	0.007634	3379
11	12	0.000442	3707	8	0.009168	3715
12	13	6.86E−05	4038	9	0.0012	4050
13	14	4.78E−05	4368	10	0.000521	4384
14	15	3.85E−05	4697	11	0.000473	4717
15	16	1.04E−05	5025			
16	17	7.49E−06	5352			
17	18	6.9E−07	5678			
18	19	1.68E−08	6003			

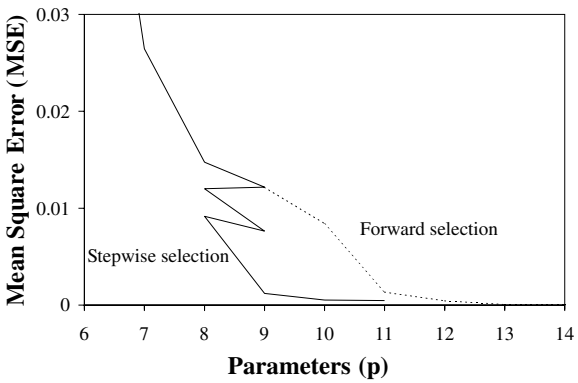

Fig. 1. Stepwise regression solutions for example 1.

selection procedure continues to add terms to the expression as shown by the dotted line in Fig. 1 because neither of the two insignificant terms was the last to enter the model. Forward selection continues to add terms to the model until the limit of 19 terms is reached without finding Eq. (2). Beyond this point the value of MSE is undefined and beyond 20 terms the solution by least squares is underdetermined, resulting in a singular matrix. Stepwise selection removes the

least significant of the two terms in the nine term expression. In successive iterations stepwise selection produces two alternative eight and nine term expressions with improved MSE values, then continues to add terms until it creates an 11 term expression for which the last term to enter is an insignificant term. To prevent cycling the algorithm must terminate at this point without discovering the correct solution.

Fig. 1 shows that the best-first search strategy of forward selection overlooks the better solutions that stepwise selection finds in the range between 8 and 11 terms. Through the process of eliminating insignificant terms stepwise selection is able to explore more combinations of terms thereby producing better expressions without increasing the length of the expressions produced. However, the process of exploring combinations of terms does not continue long enough to discover the correct form of the equation. In contrast to the two stepwise regression methods the evolution-based operations of crossover and mutation used by symbolic regression and the hybrid method can explore combinations of terms with a strategy that does not necessarily terminate at any time.

It is important to recognise that this example represents a relatively simple problem for stepwise regression to solve. If all the three key terms in the target expression are included in any of the expressions the coefficients of the non-target terms will assume values of zero and be removed in the subsequent iterations of elimination that will follow. If two of the three terms in the target expression are included in any expression the third term is guaranteed to be included in the next iteration of the selection procedure. One of the three target terms, $x_1^2x_2^2x_3$ is included from the first iteration. The stepwise regression procedure is only required to find one more target term before reaching the limits on the length of expressions imposed by the number of data points or terminating due to the significance of the last term included. Selecting terms at random has a 9.18% chance of producing the correct equation.

3.4. The results of symbolic regression

The symbolic regression method required two separate data sets, a training set and a validation set. The training set consisted of the same 20 data points used by the hybrid method and stepwise regression. The validation set consisted of an additional 20 points generated using Eq. (2). The results reported in Table 4 are taken from the best-of-run solution with respect to the training data.

Genetic programming software typically requires the user to specify a large number of parameters. For this problem the Discipulus Pro software was used with six different settings in 60 trials. In all 60 runs the population size was set at 500 and the search consisted of 100,000 tournaments (equivalent to 200 generations). Thirty of the trials used a subset of the available operators suitable for creating polynomials consisting of addition, subtraction, multi-

Table 4
MSE of symbolic regression models for example 1 (60 trials)

	3 constants		6 constants		12 constants	
	Polynomial	Non-polynomial	Polynomial	Non-polynomial	Polynomial	Non-polynomial
Maximum	13.2000	5.3142	5.7642	4.4838	19.8708	6.0032
Minimum	2.6170	1.5281	1.4198	0.9313	1.3570	1.5604
Mean	6.5474	3.0950	3.0883	2.6609	5.6587	3.0419

plication and division. The columns in Table 4 designated 'polynomial' refer to trials using the subset of operators. For the remaining 30 trials, designated 'non-polynomial' in Table 4, the set of operators included all available mathematical functions including transcendental functions. For this problem performance was not significantly affected by the choice of operators. The number of constants in Table 4 refers to the number of unique ephemeral random constants. The number of ephemeral random constants had little effect on the error.

Symbolic regression did not find the target expression in any of the trials. According to Table 4 all of the solutions produced by stepwise regression with four or more terms were better than the best results obtained with symbolic regression.

3.5. Random error and overfitting

This section examines the effects of random error in the response variable and the related problem of overfitting. The tests in this section demonstrate how error resulting from overfitting can be avoided when using the hybrid method. The two test cases are based on Eq. (2) with varying degrees of random error applied to the response variable, y.

In the first case the random error is relatively small and the onset of overfitting is easily identified from a graph of the trade-off curve between MSE and the number of parameters. Fig. 2 is a graph similar to Fig. 1. In this test Eq. (2) was used to create the response value, y, for 200 data points. Normally distributed random deviates with a variance, σ^2, equal to 2.589 were added to the response variable to simulate random noise. The standard deviation of the error, 1.609, corresponds to approximately 5% of the mean response value, $\bar{y}$. The solutions in Table 5 and the trade-off curve in Fig. 2 were found after 4000 evaluations at which point the MSE values of the best solutions appeared to have stabilised with no appreciable improvement. The program used the following configuration parameters: a maximum power value of 6, a maximum expression length of eight terms, and a maximum population size of 40.

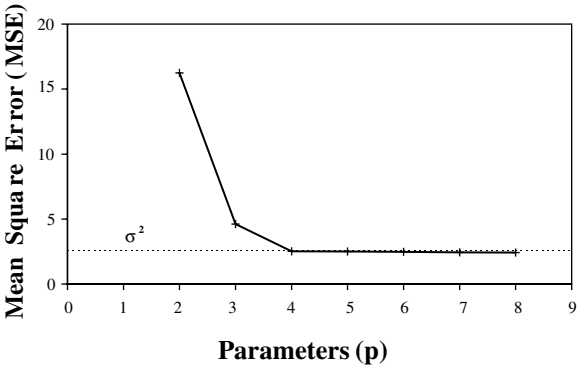

Fig. 2. MSE and number of parameters with 5% random error.

Table 5
Best solutions with 5% error

P	MSE	C_{hp}
2	16.244	1009.678
3	4.613	158.130
4	2.524	10.805
5	2.508	15.459
6	2.480	19.208
7	2.432	21.532
8	2.425	26.831

In Fig. 2 a horizontal segment appears in the trade-off curve at $p > 4$. MSE values to the right of $p = 4$ have approximately the same MSE value, which is slightly lower than 2.589, the variance of the random error, shown as the dashed line. Although the total sum of squared errors (RSS) for the best solutions decreases monotonically with p for models with $p > 4$ the MSE value tends to remain nearly constant at a value near the variance of the random error, σ^2. A horizontal segment on the right side of the trade-off curve is one indication of the onset of overfitting. Another indication is the fact that all the models with $p > 4$ contain the same three key terms found in the four term model. Simple observations of this type work in the case where the underlying trend is polynomial and the error is relatively small. A more rigorous approach is required when the error is large or the trend is transcendental. Either condition can create a smooth transition to overfitting where the best model is not obvious from the trade-off curve. The next example illustrates the use of the C_{hp} statistic. C_{hp} is derived from Mallows's C_p statistic, which is a method for comparing different sized models on the basis of relative prediction error (Table 6).

Table 6
Error in four sets of validation data for 5% error

P	RSS			
	Trial 1	Trial 2	Trial 3	Trial 4
2	3198.16	3595.13	3388.33	3717.98
3	779.00	829.86	786.91	901.04
4	377.50	392.80	383.43	548.41
5	385.14	395.57	391.29	542.50
6	391.09	404.71	396.13	548.01
7	410.89	440.45	413.63	633.67
8	393.78	418.74	397.95	596.65

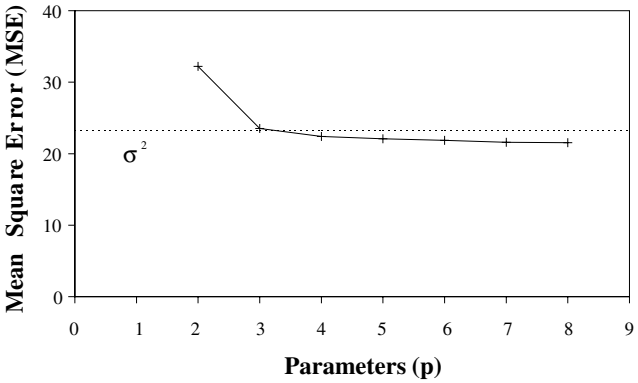

Fig. 3. MSE and number of parameters with 15% random error.

Table 7
Best solutions with 15% error

P	MSE	C_{hp}
2	32.224	80.618
3	23.517	14.145
4	22.399	10.687
5	22.095	13.974
6	21.872	17.949
7	21.588	21.444
8	21.536	26.831

The solutions in Fig. 3 and Table 7 are derived from the same data set as Fig. 2 except that the standard deviation has been increased to 4.827, approximately 15% of the mean response. In Fig. 3 a similar transition to a horizontal line appears at the three term model and the MSE value of this model is nearly equal to the variance of the random error, 23.302, shown as the

Table 8
Error in four sets of validation data for 15% error

P	RSS			
	Trial 1	Trial 2	Trial 3	Trial 4
2	5575.54	6695.12	6087.47	7996.50
3	3868.09	4068.26	3890.25	5084.15
4	3457.45	3715.03	3616.88	5069.48
5	3523.22	3733.50	3691.15	5096.91
6	3563.78	3827.19	3678.92	5319.79
7	3655.05	3763.02	3839.30	5164.24
8	3617.64	3822.53	3830.48	5325.88

dashed line in Fig. 3. Given the previous conclusions, the graph in Fig. 3 appears to indicate that the three term model is optimal and that models with four or more terms are overfit. However, on the basis of four trials with validation data (see Table 8) the four term model consistently provides the best predictions. Simple observations of the MSE values are not the best method for selecting the optimal model.

The Mallows's C_p statistic represents a method to select the model with the lowest prediction error. Overfitting will produce increased prediction error as will lack of fit (bias error). The model with the lowest prediction error will represent the best trade-off between the elimination of bias error and complexity. Eq. (5) is the formula for C_p:

$$C_p = \frac{\mathrm{RSS}_p}{\mathrm{MSE}^*} - n + 2p \tag{5}$$

where RSS_p is the residual sum of squares for the expression with p terms; MSE^* is an estimate of the variance of the random error, σ^2; n is the number of data points in the data set; and p is the number of adjustable parameters in the model.

Previous work undertaken on the C_p statistic (Refs. [12–15]) has considered first order models. These studies enumerate and compare all possible first order models including one model, a complete model that contains all variables. To calculate C_p, MSE^* is normally the MSE of the complete model. For the hybrid method a complete model consisting of all possible transformed variables is likely to be underdetermined, so MSE^* is assumed to be the MSE value of the largest model, $\mathrm{MSE}_{p_{\max}}$, where $p_{\max}$ is the number of terms in the largest expression.

To obtain an accurate estimate of σ^2, the best model with $p_{\max}$ terms can be an overfit model, but all bias error should be removed. Experience has shown that the MSE of overfit unbiased models produced by the hybrid method tends to slightly underestimate σ^2, as the graphs in Figs. 2 and 3 indicate the un-

derestimation is not due to the process of overfitting itself, but rather it is due to the ability of the evolutionary search method to find better than average patterns in noise. The calculation of C_p is sensitive to any underestimation of σ^2. Central to the derivation of the C_p statistic is the concept that the total squared error, RSS_p, is expected to decrease by σ^2 with every term added to an unbiased model. It is true for overfit models selected at random that the mean reduction in RSS_p is σ^2 with each new term added. However, the hybrid method does not produce models by random selection and the models selected to form the trade-off curve have the lowest MSE value encountered, not the mean MSE value. Therefore, the reduction in RSS_p per term is substantially greater than σ^2. The new statistic, C_{hp}, is derived by estimating C_p for the worst case scenario in which the reduction in RSS_p expected for an already overfit model is $h\sigma^2$ per new term added, where h is substantially greater than 1.

$$C_{hp} = \frac{\mathrm{RSS}_p}{\left(\frac{n-p_{\max}}{n-hp_{\max}}\right)\mathrm{MSE}_{p_{\max}}} - n + 2hp \tag{6}$$

The value of h is derived empirically by measuring the ability of the hybrid method to find false trends in random noise typical of that contained in the data set. The first step in estimating h is to extract the residual error, e_i, of the best model with $p_{\max}$ terms. Eq. (7) is the formula for the residual error, e_i:

$$e_i = y_i - \hat{y}_i \tag{7}$$

where y_i is the response for the ith observation; and $\hat{y}_i$ is the predicted response for the ith observation (predicted using the best model with $p_{\max}$ terms).

A new set of models is created with the same predictor variable values in the data set and program configuration parameters. However, the response variable in the new data set consists of the residual error extracted from the model with $p_{\max}$ terms. The residual errors are shuffled randomly so that individual errors, e_i, are not reassigned to predictor variables x_{jk} where j is equal to i. The random shuffling ensures that any remaining bias error that may have been missed does not influence the fit of the next set of models and thereby ensures that the new response values consist of pure random error. The procedure of fitting models to pure random noise is repeated several times using different random orderings of the residual. The highest value of h encountered during this process is then used to calculate the C_{hp} values for the original models according to Eq. (6).

Fig. 4 represents one example of the trade-off curve produced when fitting shuffled residual error. The dashed line is the initial mean square error of the residual, MSE_0, calculated by Eq. (8).

$$\mathrm{MSE}_0 = \frac{\sum_i^n e_i^2}{n} \tag{8}$$

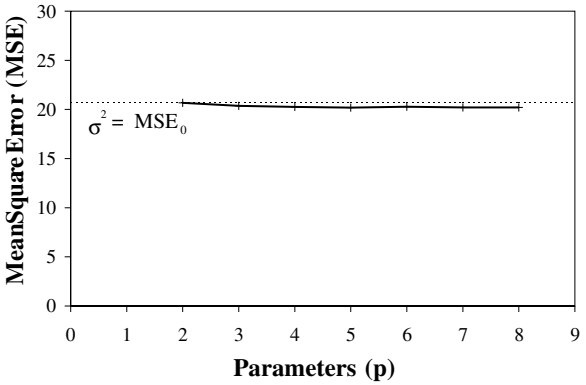

Fig. 4. MSE and number of parameters with pure error.

Table 9
Six trials with shuffled residual error

P	MSE						h_{max}
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	
2	20.665	20.665	20.665	20.184	20.187	20.196	3.354
3	20.364	20.313	20.326	20.177	20.171	20.096	2.841
4	20.261	20.307	20.261	20.123	20.019	20.133	2.556
5	20.191	20.313	20.179	20.028	19.965	19.925	2.415
6	20.270	20.036	20.231	19.897	19.888	19.684	2.550
7	20.207	20.046	20.191	19.912	19.894	19.736	2.252
8	20.205	20.086	20.018	19.934	19.847	19.652	2.188

Table 9 lists the MSE values from six trials with two different random orderings of the residual from the eight term model in Fig. 3. Column 2 lists the MSE values in Fig. 4. For each row in Table 9 the lowest MSE value is used to calculate h values in column 8 of Table 9. The MSE_0 for Table 9 is 20.675 shown as the dashed line in Fig. 4. Eq. (9) is the formula for h.

$$h = \frac{n}{p} - \frac{MSE_p(n-p)}{MSE_0 p} \tag{9}$$

The highest h value, 3.35, is used in Eq. (6) calculate the C_{hp} values in column 3 of Table 7. The low C_{hp} value of the four term model indicates that it is likely to produce better predictions than either the three and five term models. To confirm that C_{hp} selects the correct model, four new data sets were created for validation. In the new data sets different normal random deviates with the same variance, $\sigma^2 = 23.302$, have been applied to the response value. Table 8 lists the total squared error (RSS) for each of the four validation trials, in

which the models from Table 7 and Fig. 3 were used to predict the response with the new data sets. Table 8 confirms that the four term model consistently produced the best predictions.

The procedure used to calculate C_{hp} is performed on the first example in Fig. 2. For consistency the same h value, 3.35, is used again. The C_{hp} values are listed in column 3 of Table 5. The four term expression in Table 5 has the lowest C_{hp} value. The validation tests in Table 6 show that the four term expression produced better predictions that the five term expression with one exception, trial 4.

Because all of the overfit models in Table 5 (Fig. 2) are supersets of the four term model it is possible to examine whether the backward elimination procedure could identify and correct overfit models. The four term expression from Table 5 which has the same form as Eq. (2) although there is some error in estimating the parameter values and an intercept term. The five term expression contains the additional term $x_1x_2^6x_3^3$. The t-statistic for the term is 1.484 which has a corresponding p-value of 0.140. Using 95% confidence test this overfit term would be removed by backward elimination. The six term model has two additional terms, $x_1^6x_2^2x_3^6$ and $x_1^4x_2^5x_3^5$, which have t-statistics of −2.26 and 2.27 respectively. The p-values are 0.0248 and 0.0241. The terms would not be removed at the 95% confidence level but would be removed at the 99% level.

There are three additional terms in the seven term model but all appear statistically significant although the model is clearly overfit. The term with the lowest t-statistic and therefore highest p-value is $x_1x_2^3x_3^2$ which is one of the key terms in the original expression. The t-statistic was 5.483446 and the corresponding p-value was 1.3E−07. For the larger models overfit models t-statistics alone cannot be relied on to identify overfitting when the hybrid method is used.

Gorman and Toman [12] have examined the mathematical relationship between the partial F and t-statistics and Mallows's C_p. In the case where the inclusion of one additional term is considered. Mallows's C_p is less restrictive than 95% confidence. Table 10 lists the relationship between the various tests and selection criteria including C_{hp} for the value of h used in the two examples, 3.35. (The number of degrees of freedom in Table 10 is arbitrarily chosen as 200.) It should be noted that unlike t and F-statistics, C_p and C_{hp} do not require the residual error to be normally distributed with a constant variance.

Table 10
Relationship between F, t, C_p and C_{hp}

Acceptance criterion	F^*	t^*	Percent confidence
$\text{MSE}_p > \text{MSE}_{p+1}$	1	1	68.15
$C_p > C_{p+1}$	2	1.41	84.11
95% confidence	3.89	1.97	95.00
$C_{hp} > C_{hp+1}$ ($h = 3.35$)	6.7	2.59	98.97

4. Example 2: The Colebrook–White formula

This section of the paper compares the effectiveness of the three methods on a non-polynomial problem. The Colebrook–White formula, Eq. (10), calculates f, the friction factor for turbulent fluid flowing through a pipe:

$$\frac{1}{\sqrt{f}} = -2\log\left(\frac{2.51}{Re\sqrt{f}} + \frac{k}{3.7D}\right) \tag{10}$$

where Re is the Reynolds number; k is the wall roughness; and D is the diameter of the pipe.

The formula is often used in pipe network simulation software. It has an implicit form in which the value of f appears on both sides of the equation. Obtaining an accurate solution for f can be very time consuming, requiring many iterations. An approximate equation for f that does not require iteration can be used to improve the speed of simulation software.

The three methods were used to derive explicit approximators to the Colebrook–White formula for a range of Reynolds numbers and relative roughness values (k/D) where the surface of the response variable f is known to be highly non-linear. The data set consists of a two dimensional grid of 220 data points, created from twenty Reynolds values selected in equal increments of 100,000 on the interval of 100,000 to 2,000,000, and 11 relative roughness values selected in equal increments of 0.0005 on the interval of 0 to 0.005. The target values, y, for the 220 points are f values obtained using the Colebook–White formula (Eq. (10)) scaled to fit on an interval between 0 and 10 using Eq. (11).

$$y = \frac{f - 0.010373}{0.020933} \tag{11}$$

The set of transformed variables used by the stepwise regression method and the hybrid method consist of 48 terms resulting from a maximum power value of 6. The maximum length was 25 terms and the minimum significance was 95%.

Fig. 5 summarises the results obtained from the three methods. MSE values in Fig. 5 and Table 11 are based on the scaled y values of Eq. (11) and not the f values of Eq. (10). As in Fig. 1 the graph of stepwise selection solutions shows evidence of improvements created by the elimination of insignificant terms. The thin line represents the solutions generated by stepwise selection. The dotted line in Fig. 5 represents the path of solutions generated by forward selection after it diverges from stepwise selection. The bold line represents the best solutions obtained using the hybrid method. The dashed horizontal line at the bottom of the figure represents the best result obtained using symbolic regression. A horizontal line is used because the number of parameters does not directly apply to symbolic regression solutions.

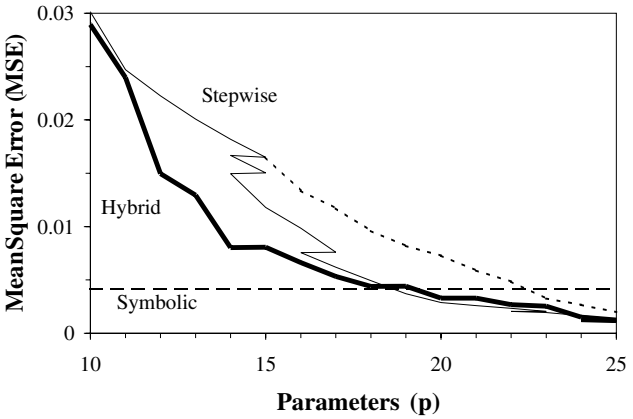

Fig. 5. Accuracy and complexity of solutions for example 2.

Table 11
MSE of symbolic regression models for example 2 (60 trials)

	3 constants		6 constants		12 constants	
	Polynomial	Non-polynomial	Polynomial	Non-polynomial	Polynomial	Non-polynomial
Maximum	0.7602	0.0895	0.6522	0.2915	0.6125	0.1204
Minimum	0.2485	0.0041	0.1799	0.0042	0.1633	0.0041
Mean	0.5804	0.0459	0.4566	0.0946	0.3264	0.0548

As in the first example symbolic regression solutions were obtained from 60 trials using Discipulus with different parameter settings following the same pattern as the first example. Table 5 shows the results of 60 trials in which 30 used transcendental functions and 30 did not. The MSE for the expressions that used transcendental functions were generally an order of magnitude more accurate than those that did not. The best MSE obtained from trials with transcendental functions is 0.0041 compared with 0.1633 for non-transcendental forms. Mean MSE values were 0.0651 for transcendental trials and 0.4545 for non-transcendental.

Fig. 5 shows that the best results obtained through symbolic regression are competitive with those produced by other methods in terms of accuracy. However, the best solutions generated by symbolic regression were very complex and included so many transcendental function calls that the computational effort was comparable to or greater than the Colebrook–White formula itself. In contrast even the largest polynomial models involve less computational effort than solutions with a single transcendental function call.

The solutions produced by the hybrid method represent a better trade-off between accuracy and computational effort than stepwise regression for expressions with 18 terms or less. Stepwise selection overtakes the hybrid method for expressions with 19 terms or more. The computational effort required to generate good solutions appears to be unaffected by the maximum length of the expression for both stepwise regression methods while the performance of the hybrid method appears to be adversely affected by the length of expressions.

5. Example 3: Rainfall runoff modelling

For this example problem stepwise regression and the hybrid method are compared with a wider variety of modelling techniques. Savic et al. [16] have reported the results of several modelling techniques to simulate rainfall runoff in the Kirkton catchment in Scotland. The methods include the conceptual model HYRROM and a more complex variation of the same program (Refs. [17,18]), an artificial neural network (Ref. [19]), and symbolic regression using a variation of the method that includes two adjustable parameters (Ref. [20]). The data set consists of daily rainfall, daily stream flow and monthly Penman open water evaporation from May 1984 to December 1988, a total of 1706 days.

The genetic programming model (Ref. [20]) and the artificial neural network model (Ref. [19]) use a set of 10 input variables to predict daily stream flow. The set of variables consists of rainfall, stream flow and evaporation for each of the three days prior to the current day as well as rainfall on the current day. The polynomial models reported in this paper use a subset of 6 of the 10 predictor variables consisting of rainfall for the current day and two previous days and stream flow from the three previous days. The conceptual models use rainfall and evaporation from the current day and create lagged inputs through a complex arrangement of calibrated internal storages and delays.

For the hybrid method the maximum length of expressions was set at 8 and the maximum power was set to 3. After the hybrid method generated 20,000 solutions the maximum length restriction was relaxed to 10 terms. Fig. 6 shows the accuracy–complexity trade-off curve generated by the hybrid method after 60,000 solutions as a bold line. The thin line represents the best solutions generated by both stepwise regression methods up to the generation of the first 10 term model. Forward selection and stepwise selection produce identical results in this case. The dashed line represents symbolic regression, the best of the four methods reported in Ref. [16]. Table 12 lists the MSE values for all the methods reported in Ref. [16] along with the stepwise and hybrid solutions. The 9 and 10 term polynomials produced by the hybrid method are the most accurate.

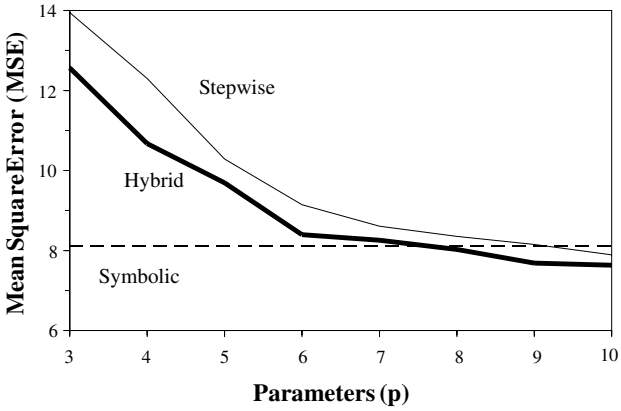

Fig. 6. Accuracy and complexity of solutions for example 3.

Table 12
Rainfall-runoff models

Previous models		MSE
Genetic program ([20])		8.102
9 parameter conceptual model ([18])		9.859
35 parameter conceptual model ([18])		9.394
Artificial neural network ([19])		8.712
Polynomial models		
Terms	Stepwise MSE	Hybrid MSE
2	17.663	17.663
3	13.945	12.565
4	12.301	10.670
5	10.286	9.691
6	9.146	8.397
7	8.609	8.256
8	8.352	8.029
9	8.154	7.685
10	7.890	7.635

6. Conclusions

The paper has described a new regression method, the hybrid method, and compared it with two established regression methods in example problems. Combining numerical parameter optimisation with the evolutionary approach of symbolic regression has required substantial modification to the original symbolic regression algorithm including new operators of crossover and mutation and a new method for generating starting populations. The first example problem demonstrated that the new approach works effectively. The new

algorithm consistently found the target expression in a very small fraction of the number of solutions expected from least-squares optimisation of randomly generated equation forms. Both the stepwise regression methods and symbolic regression failed to find the target equation at all.

The first example problem illustrated the advantage of the extended exploration of new combinations of terms when comparing the effectiveness of stepwise selection and forward selection. However, the extended exploration offered by stepwise selection is relatively restricted. In contrast the evolutionary approach used in symbolic regression and the hybrid method is interminable. In practical terms, however, symbolic regression searches stagnate due to the problem of code bloat. The hybrid method is the only approach of the three capable of an efficient and interminable search of combinations of terms.

It is not surprising that symbolic regression produced the worst performance of the three methods in first example since the other two methods were specifically designed for polynomial problems. The stepwise and hybrid methods continued to outperform symbolic regression in the other two example problems using relatively small polynomial functions. Clearly for these two problems the cost of using non-adjustable parameters is greater than the benefits resulting form the wide variety of operators and equation forms available to symbolic regression.

In comparing the hybrid and stepwise regression methods, the hybrid method produced more accurate models than stepwise regression for all sizes of expressions in the rainfall-runoff problem and for the shorter length expressions in the Colebrook–White approximation problem. Computational effort associated with the hybrid method appears to increase with increased expression length unlike the stepwise regression methods, but the hybrid method does produce more accurate short-length expressions. Future work will be directed at improving the efficiency of search of the hybrid method for large expressions.

Acknowledgement

This work was supported by the UK Engineering and Physical Sciences Research Council, grant GR/L67189.

References

[1] J.R. Koza, Genetic Programming: On the Programming of Computers by means of Natural Selection, MIT Press, Cambridge, MA, 1992.

[2] B. McKay, M.J. Willis, G.W. Barton, Using a tree structured genetic algorithm to perform symbolic regression, in: Genetic Algorithms in Engineering Systems: Innovations and Applications, IEE, Conference Publication no. 414, 1995, pp. 487–492.

[3] A. Watson, I. Parmee. Systems identification using genetic programming, in: Proceedings of ACEDC '96 PEDC, University of Plymouth, UK, 1996.
[4] W.B. Langdon, T. Soule, R. Poli, J.A. Foster, The evolution of size and shape, in: L. Spector, W.B. Langdon, U.M. O'Reilly, P.J. Angeline (Eds.), Advances in Genetic Programming III, MIT Press, Cambridge, 1999, pp. 163–190.
[5] P.W.H. Smith, Controlling code growth in genetic programming, in: R. John, R. Birkenhead (Eds.), Advances in Soft Computing: Soft Computing Techniques and Applications, Physica-Verlag, Heidelberg, 1999, pp. 166–171.
[6] V. Babovic, M. Keijzer, Genetic programming as a model induction engine, Journal of Hydroinformatics 2 (1) (2000) 35–61.
[7] M. Keijzer, V. Babovic, Dimensionally aware genetic programming, in: W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, R.E. Smith (Eds.), GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann, San Francisco, 1999.
[8] J.W. Davidson, D.A. Savic, G.A. Walters, Symbolic and numerical regression: a hybrid technique for polynomial approximators, in: R. John, R. Birkenhead (Eds.), Advances in Soft Computing: Soft Computing Techniques and Applications, Physica-Verlag, Heidelberg, 1999, pp. 111–116.
[9] N.R. Draper, H. Smith, Applied Regression Analysis, John Wiley and Sons, New York, 1998.
[10] F.D. Francone, Discipulus Pro owner's manual, Register Machine Learning Technologies Inc., Oakland, 1998.
[11] H.O. Poyhonen, D.A. Savic, Symbolic Regression using Object-oriented Genetic Programming (in C++): Centre for Water Systems and Control Engineering, Report no. 96/04, University of Exeter, Exeter, UK, 1996.
[12] J.W. Gorman, R.J. Toman, Selection of variables for fitting equations to data, Technometrics 8 (1) (1966) 27–51.
[13] C.L. Mallows, Some comments on C_p, Technometrics 15 (4) (1973) 661–675.
[14] C.L. Mallows, More comments on C_p, Technometrics 37 (4) (1995) 362–372.
[15] S.G. Gilmore, The interpretation of Mallows's C_p-statistic, The Statistician 45 (1) (1996) 49–56.
[16] D.A. Savic, G.A. Walters, J.W. Davidson, A genetic programming approach to rainfall-runoff modelling, Water Resources Management 13 (1999) 219–231.
[17] C.W.O. Eeles, Y. Parks, A. Barr, HYRROM Operation Manual Institute of Hydrology, Wallingford, Oxfordshire, UK, 1989.
[18] C.W.O. Eeles, Parameter Optimization of Conceptual Hydrological Models Ph.D. Thesis, Open University, Milton Keynes, UK, 1994.
[19] F. Jacq, D.A. Savic, Rainfall-runoff Modelling using Neural Networks Centre for Water Systems And Control Engineering, Report no. 97/02, School of Engineering, University of Exeter, Exeter, UK, 1997.
[20] N. Cousin, D.A. Savic, A Rainfall-runoff Model using Genetic Programming Centre for Water Systems And Control Engineering, Report no. 97/03, School of Engineering, University of Exeter, Exeter, UK, 1997.