We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter 4

Genetically Programmed Regression Linear Models for Non-Deterministic Estimates

Guilherme Esmeraldo, Robson Feitosa,
Dilza Esmeraldo and Edna Barros

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/48156

1. Introduction

Symbolic regression is a technique which characterizes, through mathematical functions, response variables with basis on input variables. Their main features include: need for no (or just a few) assumptions about the mathematical model; the coverage of multidimensional data, frequently unbalanced with big or small samples. In order to find the plausible Symbolic Regression Models (SRM), we used the genetic programming (GP) technique [1].

Genetic programming (GP) is a specialization of genetic algorithms (GA), an evolutionary algorithm-based methodology inspired by biological evolution, to find predictive functions. Each GP individual is evaluated by performing its function in order to determine how its output fits to the desired output [2,3].

However, depending on the problem, one may notice that the estimates of the SRM found from the GP may present errors [4], affecting the precision of the predictive function. To deal with this problem, some studies [5,6] substitute the predictive functions, which are deterministic mathematical models, by linear regression statistical models (LRM) to compose the genetic individual models.

LRM, as well as the traditional mathematical models, can be used to model a problem and make estimates. Their great advantage is the possibility of controlling the estimate errors. Nevertheless, the studies available in the literature [5,6] have considered only information criteria, such as the sum of least squares [7] and AIC [8], as evaluation indexes with respect to the dataset and comparison of the solution candidate models. Despite the models obtained through this technique generate good indexes, sometimes the final models may not be representative, since the model structure assumptions were not verified, bringing some incorrect estimates [9].

INTECH
open science | open minds

© 2012 Esmeraldo et al., licensee InTech. This is an open access chapter distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

So, in this study we propose the use of statistical inference and residual analysis to evaluate the final model, obtained through GP, where we check the assumptions about the structure of the model. In order to evaluate the proposed approach, we carried out some experiments with the prediction of performance of applications in embedded systems.

This chapter is organized as follows. In Section 2, we briefly introduce the theoretical basis of the regression analysis. In Section 3, we detail the main points of the proposed approach. In Section 4, we introduce the application of the proposed approach through a case study. Section 5 shows the experimental results of the case study. Finally, in Section 6, we raise the conclusions obtained with this work.

2. Linear regression background

Like most of the statistical analysis techniques, the objective of the linear regression analysis is to summarize, through a mathematical model called Linear Regression Model (LRM), the relations among variables in a simple and useful way [10]. In some problems, they can also be used to specify how one of the variables, in this case called response variable or dependent variable, varies as a function of the change in the values of the other variables of the relation, called predictive variables, regressive variables or systematic variables.

The predictive variables can be quantitative or qualitative. The quantitative variables are those which can be measured through a quantitative scale (i.e., they have a measurement unit). On the other hand, the qualitative variables are divided in classes. The individual classes of a classification are called *levels* or *classes* of a factor. In the classification of data in terms of factors and levels, the important characteristic that is observed is the extent of the variables of a factor which can influence the variable of interest [11]. These factors are often represented by dummy variables [12].

Let D be a factor with five levels. The jth dummy variable U_j for the factor D, with $j=1,...,5$, has the ith value u_{ij}, for $i=1,...,n$, given by

$$u_{ij} = \begin{cases} 1, & if \quad D_i = j^{th} \quad category \quad of \quad D \\ 0, & otherwise \end{cases} . \tag{1}$$

For instance, let there be a variable, which supports a certain characteristic x, as a two-level factor D. Taking a sample, shown in Table 1, with 5 different configurations, we can represent the factor D with the dummy variables of Table 2.

	Support to characteristic *x*
1	Yes
2	No
3	Yes
4	No
5	No

Table 1. Sample with size 5, with several pipeline support configurations.

	u1	u2
1	1	0
2	0	1
3	1	0
4	0	1
5	0	1

Table 2. Representation of the sample of Table 1, through dummy variables.

We can see in Table 2 that the configurations with support to the characteristic *x* had values *u1=1* and *u2=0*, and that the configurations without support had values *u1=0* and *u2=1*.

LRMs may also consider the combination of two or more factors. When the LRM has more than one factor, the effect of the combination of two or more factors is called *interaction effect*. Interactions occur when the effect of a factor varies according to the level of another factor [10]. In contrast, the effect of a simple factor, that is, without interaction, is called *main effect*. The interaction concept is given as follows: if the change in the mean of the response variable between two levels of a factor A is the same for different levels of a factor B, then we can say that there is no interaction; but if the change is different for different levels of B, then we say that there is interaction. Interactions report the effect that factors have over the risk of the model, and which are not reported in the analysis of correlation between the factors.

So, considering the relations between the dependent variables and the predictive variables, the statistical linear regression model will be comprised of two functions, one for the mean and another for the variance, defined by the following equations, respectively:

$$E(Y \mid X = x) = \beta_0 + \beta_1 x \tag{2}$$

$$Var(Y \mid X = x) = \sigma^2 \tag{3}$$

where the parameters in the *mean* function are the intercept β_0, which is the value of the mean $E(Y \mid X=x)$ when x is equal to zero, and the slope β_1, which is the rate of change in $E(Y \mid X=x)$ for a change of values of X, as we can see in Figure 1. Varying these parameters, it is possible to obtain all the line equations. In most applications, these parameters are unknown and must be estimated with basis on the problem data. So, we assume that the *variance* function is constant, with a positive value σ^2 which is normally unknown.

Differently from the mathematical models, which are deterministic, linear regression models consider the errors between the observed values and these estimated by the line equation. So, due to the variance $\sigma^2>0$, the values observed for the i^{th} response y_i are typically different from the expected values $E(Y \mid X=x_i)$. In order to consider the error between the observed and the expected data, we have the concept of statistical error, or e_i, for the case i implicitly defined by the equation:

$$y_i = E(Y \mid X = x_i) + e_i \tag{4}$$

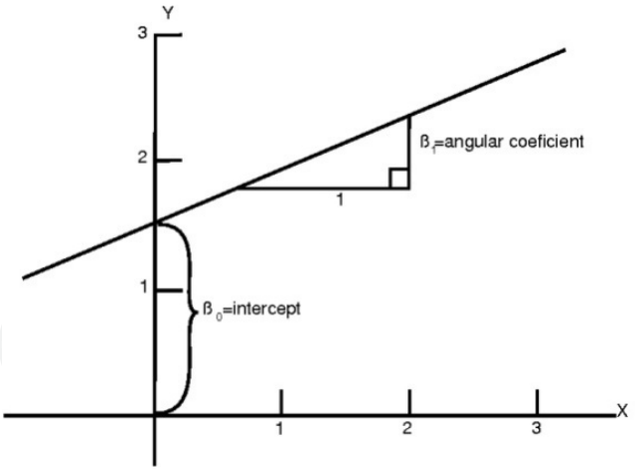


Figure 1. Graphic of the line equation $E(Y \mid X=x)=\beta_0 + \beta_1 x$.

or explicitly by:

$$e_i = y_i - E(Y \mid X = x_i) \tag{5}$$

The e_i errors depend on the unknown parameters of the *mean* function and are random variables, corresponding to the vertical distance between the point y_i and the function of the mean $E(Y \mid X=x_i)$.

We make two important assumptions about the nature of the errors. First, we assume that $E(e_i \mid x_i)=0$. The second assumption is that the errors must be independent, which means that the value of the error for one case does not generate information about the value of the error for another case. In general, we assume that the errors are normally distributed (statistical Gaussian distribution), with mean zero and variance σ^2, which is unknown.

Assuming n pairs of observations (x_1, y_1), (x_2, y_2), ..., (x_n, y_n), the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ of β_0 and β_1, respectively, must result in a line that best fits to the points. Many statistical methods are suggested to obtain estimates of the parameters of a model. Among these models, we can highlight the Least Squares and Maximum Likelihood methods. The first one stands out for being the most used estimator [13]. So, the Least Squares methods is intended to minimize the sum of the squares of the residuals e_i, which will be defined next, where the estimators are given by the equations:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} y_i x_i - \frac{(\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i)}{n}}{\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}} \tag{6}$$

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x} \tag{7}$$

where $\overline{x}$ and $\overline{y}$ are given by:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{8}$$

$$\overline{y} = \frac{\sum_{i=1}^{n} y_i}{n} \tag{9}$$

With the estimators, the regression line (or model) is given by:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{10}$$

where each pair of observations meets the relation:

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + e_i, \qquad for \quad i = 1,2,..,n \tag{11}$$

From the above equation, we can then define the residual as:

$$r = \hat{e}_i = y_i - \hat{y}_i \tag{12}$$

where $\hat{e}_i$ is the error in the fitness of the model for the ith observation of y_i.

The residuals $\hat{e}_i$ are used to obtain an estimate of the variance σ^2 through the sum of the squares of $\hat{e}_i$:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} \hat{e}_i^2}{n-2} \tag{13}$$

According to [14], the traditional project flow for modeling through LRMs can be divided into three stages: (i) formulation of models; (ii) fitness and (iii) inference.

LRMs are a very useful tool, since they are very flexible in stage (i), are simply computable in (ii) and have reasonable criteria in (iii). These stages are performed in this sequence. In the analysis of complex data, after the inference stage, we may go back to stage (i) and choose other models with basis on more detailed information obtained from (iii).

The first stage, formulation of models, covers the choice of options for the distribution of probabilities of the response variable (random component), predictive variables and the function that links these two components. The response variable used in this work consists in the estimate of the performance of the communication structure of the platform. The predictive variables are the configuration parameters of the buses contained in the space of the communication project. For this study, we analyzed several linking functions, and empirically chose the *identity* function, because it represents the direct mapping between bus configurations and their respective estimated performances.

The fitness stage consists in the process of estimation of the linear parameters of the generalized linear models. Several methods can be used to estimate the LRM parameters, such as the Least Squares and Maximum Likelihood methods.

Finally, the inference stage has the main objective of checking the adequateness of the model and performing a detailed study about the unconformities between the observations and the estimates given by the model. These unconformities, when significant, may imply in the choice of another linear model, or in the acceptance of aberrant data. Anyway, the whole methodology will have to be repeated. The analyst, in this stage, must check the precision and the interdependence of the performance estimates, build trust regions and tests about the parameters of interest, statistically analyze the residuals and make predictions.

3. Description of the proposed approach

The GP algorithm herein used follows the same guidelines of the traditional GP approaches: representation of solutions as genetic individuals; selection of the training set; generation of the starting population of genetic individuals that are solution candidates; fitness of the solution candidates to the training set; selection of parents; evolution, through selection, crossover and mutation operators [2]. Besides these activities, this work includes two new stages, which consist in the evaluation of the final model, as shown in the flow of Figure 1.

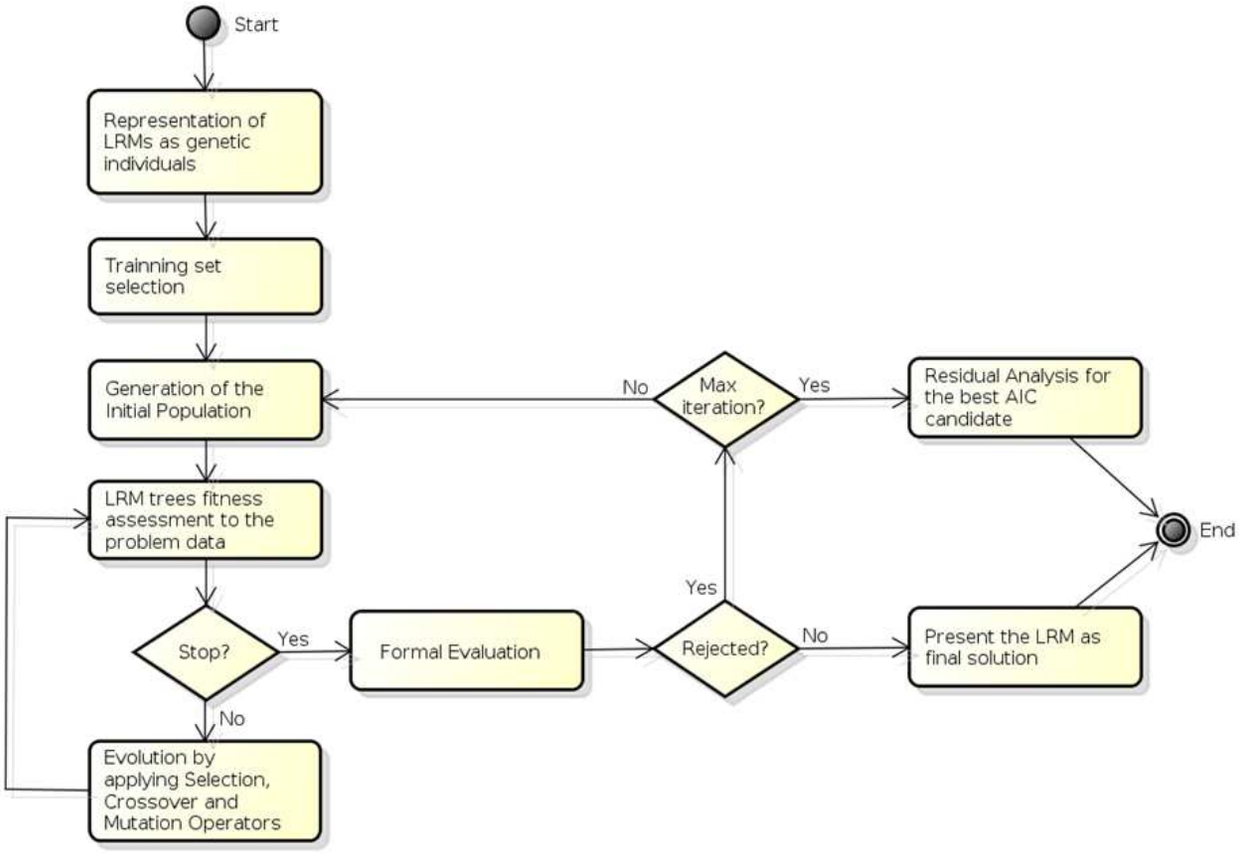


Figure 2. Flow of the proposed PG approach with LRM.

When the processing of the GP algorithms ends, due to some stop criterion, (e.g. the maximum number of generations is reached), the fittest genetic individual to the data is selected to be formally evaluated through statistical inference, with the application of the test of assumptions. Depending on the result of the evaluation, the GP algorithm can either start a new iteration, generating a new starting population, or present the LRM as a final solution.

If no candidate is approved in the formal evaluation, at the end of the iterations (limited to a maximum number as the second stop criterion), the best candidate among all the iterations may be reevaluated through residual diagnosing. In this other evaluation method, the assumptions about the model may be less formal, becoming, this way, a more subjective kind of analysis.

Each one of the activities presented in the Flow of Figure 1 will be detailed in the next subsections.

3.1. Representation of solutions as genetic individuals

GP normally uses trees as data structures [15] because the solutions are, commonly, mathematical expressions, and then it is necessary to keep their syntactic structure (trees are largely used to represent syntactic structures, defined according to some formal grammar [16]).

As seen in the previous subsection, linear regression models are statistical models comprised of two elements: a response variable and the independent variables. So, these models are structured, in the proposed approach, also as trees, called *expression trees*, where the internal nodes are either linking operators (represented by the arithmetic operator of addition) or iteration operators (represented by the arithmetic operator of multiplication) acting between the predictive variables, which are located in the leaves of the tree, as shown in Figure 3.

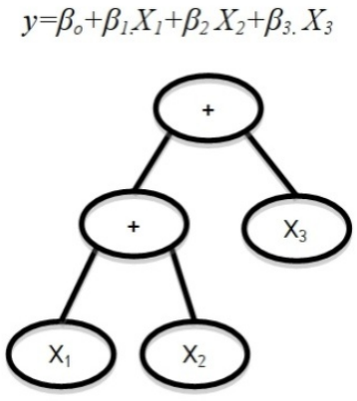


Figure 3. Example of LRM modeled as a genetic individual.

It can be seen, in the top of Figure 3, an LRM, and right below, the respective model in the form of a tree, which is the structure of a genetic individual. In this individual, we have, in

the roots of the tree and of the sub-tree in the left, the linking operator, and in the leaves we have the predictive variables X_1, X_2 and X_3.

Formally, an LRM modeled as a genetic individual can be defined as a tree containing a finite set of one or more nodes, where:

i. there is a special node called *root*.
ii. the rest of the nodes form:
    1. two distinct sets where
    2. each one of these sets is also a tree which, in this case, is also called *sub-tree*. The sub-trees may be either left or right.
iii. the roots of the tree, and of the adjacent sub-trees, is either a linking or an iteration operator.
iv. the leaves are independent variables.

Once we define the data structure that will be used to represent the LRMs as genetic individuals, the next task, as defined in the flow of Figure 2, is the selection of the points of the project space that will be used to form the training set for the GP algorithm. The following subsection gives more details about the technique chosen to select points.

3.2. Selection of the training set

The selection of the elements that will compose the training set can be done in many ways, but techniques like random sampling do not guarantee a distributed sample, and variance-based sampling does not allow to collect the whole dataset of the sample, and then the selected set may not be enough to obtain a linear regression model which enables accurate estimates. So, in this work, we use the Design of Experiment technique [17] for the selection of points that will compose the training space.

Design of experiments, also known in statistics as Controlled Experiment, refers to the process of planning, designing and analyzing an experiment so that valid and objective conclusions can be extracted effectively and efficiently. In general, these techniques are used to collect the maximum of relevant information with the minimum consumption of time and resources, and to obtain optimal solutions, even when it is impossible to have a functional mathematical (deterministic) model [17-20]

The design of experiment technique adopted in this work is known as *Audze-Eglais Uniform Latin Hypercube* [21,22]. The Audze-Eglais method is based on the following analogy to Physics:

> *Assume a system composed of points of mass unit which exert repulsive forces among each other, causing the system to have potential energy. When the points are freed, from a starting state, they move. These points will achieve equilibrium when the potential energy of the repulsive forces of the masses is minimal. If the magnitude of the repulsive forces is inversely proportional to the square of the distance between the points, then the minimization of equation below will produce a system of distributed points, as uniform as possible.*

$$U = \sum_{p=1}^{P} \sum_{q=p+1}^{P} \frac{1}{L_{pq}^2} \tag{14}$$

where U is the potential energy and is the distance between the points p and q, and $p \neq q$.

The points of the project space are comprised of the parameters of the system to be modeled, and each point is a combination of the values that these parameters can receive. The Audze-Eglais method can be applied to these project spaces, provided that we consider the intervals (the distances) between the values of each parameter of the system, and that these values are taken together, in order to minimize the objective function.

The minimization of the above equation can be performed through some optimization technique or by verification of every possible combination. The use of the second approach may be unviable, since the search for each possible combination in project spaces with many points has a high computational cost. So, in this study, we used the GPRSKit [23] tool, which uses genetic programming techniques to minimize the equation, and outputs the points of the project space identified in the optimization of the equation.

Once defined the training set, the next task is the generation of a starting population of genetic individuals, which are LRMs candidate to solution, so the genetic algorithm can evolve them.

3.3. Generation of the starting population of genetic individuals

There must be a starting population so that the evolution algorithm can act, through the application of the selection, crossover and evolution operators. For this, aiming at the variability of individuals and consequent improvement on the precision of results, we adopted the *Ramped Half-and-Half* [24] technique.

This technique selects, initially, a random value to be the maximum depth of the tree to be generated. Next, the method for generation of the new tree is selected. *Ramped Half-and-Half* uses two generation methods, where each one generates half of the population. They are described below:

- **Growing**: this method creates new trees of several sizes and shapes, regarding the depth limit previously defined. Figure 4(a) shows an example of a tree created with the application of this method. In it, we see that the leaves have different depths.
- **Complete**: a tree created with this method has its leaves with the same depth, which is also selected at random, but respects the depth limit initially selected. Figure 4(b) shows a tree created with this method. Notice that all leaves have the same depths.

3.4. Description of the utility function (Fitness)

The fitness of a candidate LRM is evaluated with basis on the quality of the estimates that it generates compared to the data obtained from the problem data. The quality of an LRM can be quantified through its fitness and its complexity, measured, in this study, by the *Akaike Information Criterion* (AIC) [8], since it is one of the most used criteria [10].

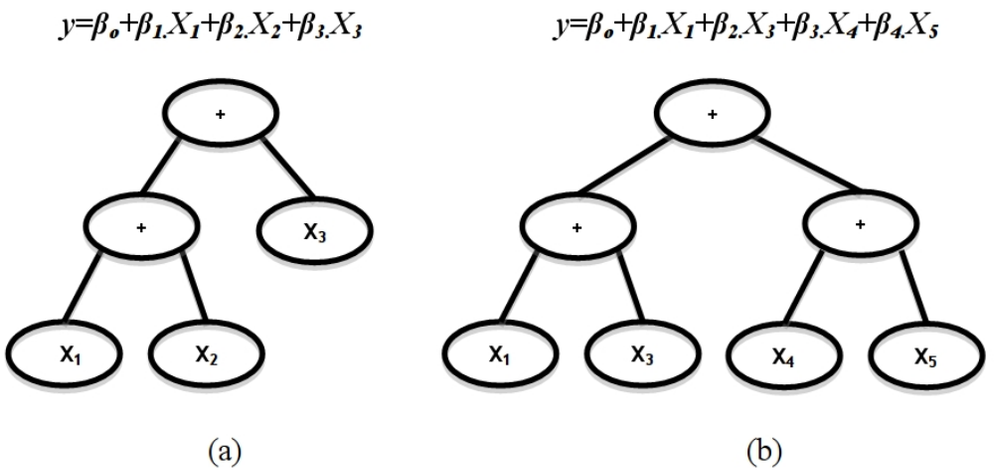


Figure 4. Examples of trees generated from (a) complete generation method and (b) generation by growing.

The AIC can be given by the following equation:

$$AIC = 2.tc - 2.\ln(L) \tag{15}$$

where tc is the number of terms of the model and L is the likeliness, which is the pooled density of all the observations. Considering an independent variable with normal distribution with mean $\beta_0 + \beta_1 x_i$ and variance σ^2, the likeliness can be given by:

$$L(\beta_0, \beta_1, \sigma^2) = \frac{1}{\sigma^2(\sqrt{2\pi})^n} e^{-\frac{1}{2}\cdot\frac{\sum_{i=1}^{n}(y_i-\beta_0-\beta_1 x_i)^2}{\sigma^2}} \tag{16}$$

3.5. Evolution

In this stage we apply, to the solution candidate genetic individuals, the selection, mutation and evolution operations. The first operation is responsible for the selection of individuals that will compose the set of parents. In this set, the genetic crossover function will act, so that the genetic content of each individual will be transferred to another one, generating new solution candidates. The objective is to group the best characteristics in certain individuals, forming better solutions. The mutation function will select some of the individuals to have their genetic content randomly changed, to cause genetic variability in the populations, avoiding the convergence of the algorithm to a local maximum.

The selection, crossover and mutation operations are described next.

3.5.1. Parents selection

The method for selection of parents must simulate the natural selection mechanism that acts on the biological species: the most qualified parents, those which better fits to the problem data, generate a large number of children, while the less qualified can also have descendents,

so avoiding premature genetic convergence. Consequently, we focus on individuals highly fitted, without completely discarding those individuals with very low degree of fitness.

In order to build a set of parent LRMs, we use the tournament selection method [25]. In this approach, a predetermined number of solution candidate LRMs are randomly chosen to compete against each other. With this selection technique, the best LRMs of the population will only have advantage over the worst, *i.e.*, they will only win the tournament if they are chosen. Tournament parameters, like tournament size and generations number, are dependent on the problem domain. In this work, they are described in case study section.

The proposed approach for GP also uses the technique of selection by elitism [26]. In this approach, only the individual having the best fitness function value is selected. With this, we guarantee that the results of the GP approach will always have a progressive increase at each generation.

3.5.2. Crossover and mutation

In order to find the LRM that best fits to the data obtained with communication graphs, the crossover and mutation operators are applied to the genetic individuals, the LRM trees, as shown in Figure 5. The crossover and mutation operators, in genetic programming, are similar to those present in conventional genetic algorithms.

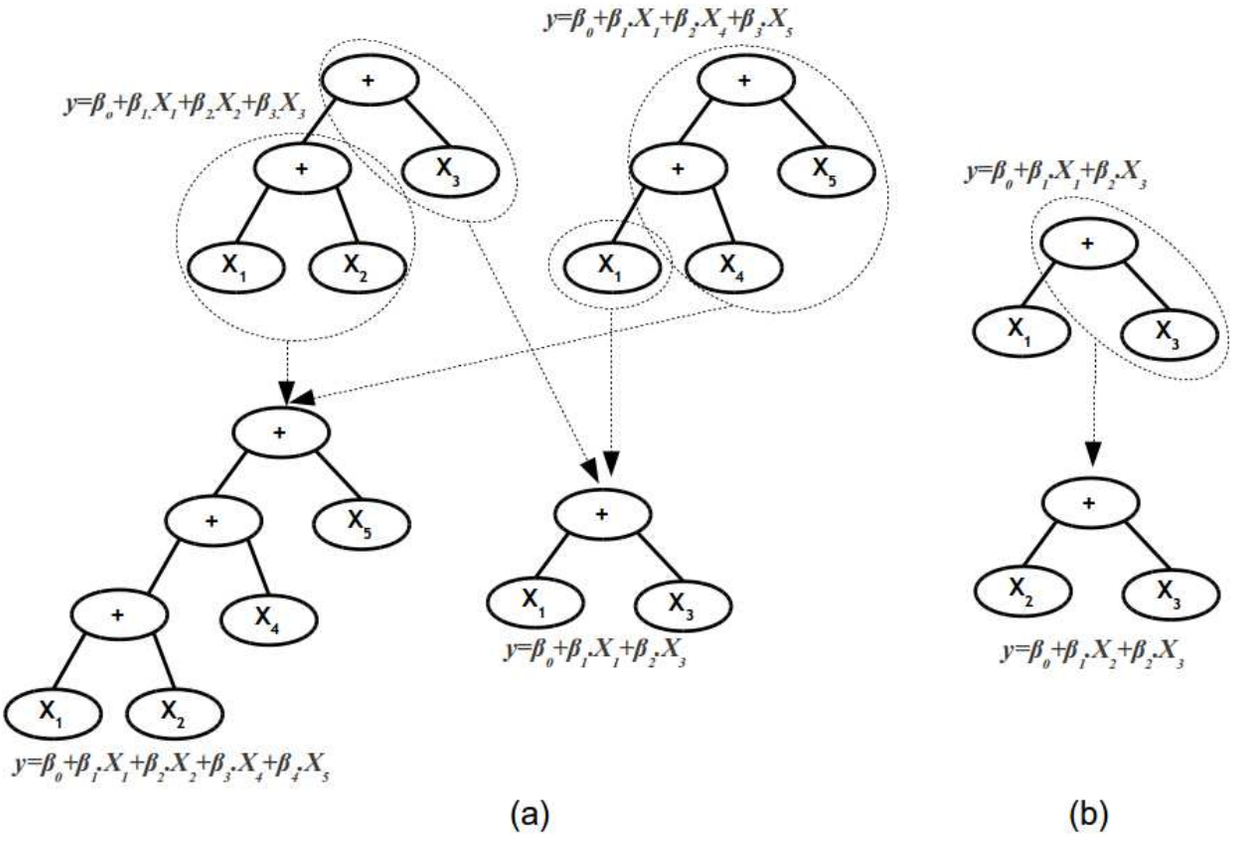


Figure 5. Expression trees representing LRMs under the operations of (a) crossover and (b) mutation.

In the first operator, represented in Figure 5 (a), the candidates are selected for reproduction according to their fitness (fittest candidates have higher probabilities of being selected) and, next, exchange their genetic content (sub-trees), randomly chosen, between each other. Figure 5(b) illustrates the crossover of the parents $y=\beta_0+ \beta_1.X_1 +\beta_2.X_2 + \beta_3.X_3$ and $y=\beta_0+ \beta_1.X_1 +\beta_2.X_4 + \beta_3.X_5$, generating the children $y=\beta_0+ \beta_1.X_1 +\beta_2.X_2 + \beta_3.X_4+ \beta_4.X_5$ and $y=\beta_0+ \beta_1.X_1 +\beta_2.X_3$.

With mutation, represented in Figure 5 (b), after a crossover operation, it is randomly generated a mutation factor for each new genetic individual. If the mutation factor exceeds a predetermined boundary, a sub-tree is selected at random in the LRM and mutated to a new different sub-tree. Figure 5 illustrates the mutation of the model $y=\beta_0+ \beta_1.X_1 + \beta_2.X_3$ to $y=\beta_0+ \beta_1.X_2 + \beta_2.X_3$, where it can be noticed that there was a mutation in the genetic content X_1 to X_2.

In the approach proposed in this work, we used the two-point crossover operator [27], because this way it combines the largest number of chromosomal schemes and, consequently, increases the performance of the technique. On the other hand, for mutation, we used the simple operator [27], because the mutation prevents the stagnation of the search with low mutation factor, but if this rate is too high, the search becomes excessively random, because the highest its value is, larger is the substituted part of the population, which may lead to the less of highly qualified structures.

3.6. Formal evaluation of a linear regression model

Once an iteration of the proposed GP algorithm is ended, the best solution found in the iteration is formally evaluated. In linear regression, assumptions about the fitted model must be considered so that the results can be reliable. So, the evaluation process consists in verifying, by residual inference, the assumptions of normality, homoscedasticity and independence about the distribution of errors of the fitted LRM. We used the following adherence tests:

- *Shapiro-Wilk* [30] to check the assumption of normality;
- *Breusch-Pagan* [31] to check the assumption homoscedasticity;
- and *Durbin-Watson* [32] to check the independence (absence of autocorrelation) among the errors.

If the result of any of these tests is not positive and the maximum number of iterations was not reached, the GP algorithm will start a new evolution iteration through the generation of a new starting population and will follow the flow presented in Figure 2. Otherwise, the algorithm presents the LRM as final solution.

3.7. Residual Analyses for the genetic individual with the best AIC

At the end of all the iterations, if no genetic individual is approved in the formal evaluations, the GP algorithm will select the solution with the best AIC for residual analysis. The residual analysis allows the evaluation of the assumptions about a model [12].

So, in this work, the residual analysis is divided in two stages:

1. Residual diagnostic plots, where we build the following diagrams:
   - Diagram of distribution of accumulated errors, to quantify the distance between the estimates given by the LRM and the data of the training set;
   - Q-Q Plots and Histograms, to check the assumptions about the error probability distributions;
   - Diagram of residuals dispersion against the fitted values, the check the assumption of homoscedasticity;
   - Diagram of dispersion of the residuals, to check the absence of autocorrelation among the errors.
2. Application of the statistical test of *Mann-Whitney-Wilcoxon* [29] to the data of the training set and the respective estimates given by the LRM found. The *Mann-Whitney-Wilcoxon* test is a non-parametric [28] statistical hypothesis test used to check whether the data of two independent sets tend to be equal (null hypothesis) or different (alternative hypothesis). With these same sets, we still perform the computation of the global mean errors, as a measurement for the central location of the set of residuals, maximums and minimums. These measurements are used to check the precision of the estimates and the possibility of presence of *outliers*.

4. Case study

In order to validate the proposed approach, we have used a case study where we predict the performance of an embedded system. The case study includes an application of the SPLASH benchmark[1] [33] for a simulation model of an embedded hardware platform. This application, which consists in the sorting a set of integers through radix [34], has two processes. The first one allocates, in a shared memory, a data structure (list), comprised of a set of integers, randomly chosen, some control flags and a *mutex* (to manage the mutually exclusive access). Once the data structure is allocated, both processes will sort the integers list, concurrently.

For the execution of the application, we designed a simulation model of a hardware platform, described in the language for modeling embedded systems, SystemC [35], comprised of two models of MIPS processors, one for each process of the application of sorting by radix, a shared memory, to stores program and application data, as well as shared data, and a ARM Amba AHB [36] shared bus model.

This model allows us to explore the bus configurations to optimize the performance of the application of radix sort.

The experiment methodology was based on the comparison between the execution times of the application, obtained by the simulation model with the estimates acquired from an LRM obtained by the proposed method. The objective is to show that the obtained models may bring highly precise estimates.

[1] Set of multiprocessed applications, used to study the following properties: computational load balance, computation rates and traffic requirements in communications, besides issues related to spatial locations and how these properties can be scalable with the size of the problems and the number of processors.

We considered the following configuration parameters for the Amba AHB bus: data bus width, fixed priority arbitrage mechanisms, operation frequency and transference types. With the combination of the possible values for these parameters, we built a project space with 72 distinct configurations.

In the representation of the LRMs, in the proposed GP algorithm, the configuration parameters of the bus ware characterized as predictive variables and the execution time of the embedded application, as the independent variable. The table below describes each one of these variables.

Variable	Representation in the LRM	Values
Data bus width	bw	8, 16, 32 (bits)
Transference type	ty	With preemption, without preemption
Operation frequency	fr	100, 166, 200 (MHz)
Priority of the first process	p1	Higher, lower (priority)
Priority of the second process	p2	Higher, lower (priority)
Execution time of the application	te	Time measured in ns

Table 3. Candidate variables to the linear regression model.

It can be seen in Table 3 that all the predictive variables have discrete values, and then they are classified as factors. In the LRMs, the predictive variables are represented as dummy variables.

With the increase in the training set, the probability of distortion on the estimates may increase, because the possibility of existence of outliers in this set may also increase. On the other hand, larger training sets may be more significant for the obtainment of a more precise model. For this reason, we used three training sets, with distinct sizes, to check these assumptions. So, we selected three sets, using the technique introduced in Subsection 3.2, with 10% (7 samples), 20% (14 samples) and 50% (36 samples) of the project space. The rest of the points were grouped in test sets, used to evaluate the precision of the estimates given by the obtained models.

According to [2], on average, 50 generations are sufficient to find an acceptable solution, and larger populations have higher probability of finding a valid solution. So, for the GP algorithm, we considered the following parameters: 1000 candidates for each generation of LRM trees; the maximum number of generations was limited in 50; and stop condition of the algorithm consisting of an LRM which is the fittest candidate for 30 consecutive generations.

For each generation, 999 tournaments were carried out, where 50 LRMs were randomly chosen to participate. During the tournament the AIC index is computed, in order to evaluate each one of the participants. So, the winners, those with the best AIC indexes, are selected for crossover. For mutation, a mutation factor is randomly computed in all the LRM trees generated by crossover. If the computed value for each tree is below 5% - index demonstrated in [37] as qualified to find good solutions in several problem types - then the three will mutate and, next, selected to make part of the next generation. Finally, the fittest LRM trees of the present

generation are automatically selected, through elitism, to complete the number of individuals of the next generation. Finally, the maximum number of iterations was limited to 50.

After the validation stages, the final models found, for the training set, had their estimates, given by prediction, compared to those of the respective training sets, as described in the next section.

5. Experimental results

As described in the previous section, we used three training sets for validation of the proposed approach. However, the application of this approach brought different results for these sets.

For the first set, that with 10% if the project space, which we will call Set A, the final model was approved in the formal evaluation, right in the first iteration. For Set B (the set with 20% of the design space), the final model was also approved in the formal evaluation, but needed five iterations. The results of the formal tests for the models selected for the Sets A and B can be seen in Table 4.

Measurement	**P-Value**		
Set	**A**	**B**	**C**
***Shapiro-Wilk* test(Normality)**	14.44%	65.69%	3.2%
***Breusch-Pagan* test (Homoscedasticity)**	53.66%	47.34%	1e-03%
***Durbin-Watson* test (Independence)**	87.2%	56.80%	82.80%

Table 4. Formal test results for verification of assumptions about the LRMs selected for the Sets A, B and C.

The test results for Sets A and B, presented in Table 4, show indexes (p-values) above the significance level, defined in this work as 5%. So, the structures of the errors of the selected LRMs, for the sets A and B, tend to have normalized errors, with constant variances and independent from each other.

Finally, for the Set C, the last training set, no model was approved in the formal evaluation. Table 4 also shows the tests results for the final model found (best AIC) for the Set C. The p-values for the *Shapiro-Wilk* and *Breusch-Pagan* tests are below the significance level, being necessary to do residual analysis. The final results of the residual analysis are shown in the graphics of Figure 6.

Figure 6 presents the graphics of (a) Q-Q Plot and (b) Residuals histograms, as well as (c) of dispersion of the values observed in the Set C versus residuals and (d) of the order of collection of residuals. Analyzing Figure 6 (b), we may notice that the errors presented by the LRM selected for the Set C do not follow a normal distribution, violating the assumption of normality of the model structure. However, it can be seen that the distribution of the errors tends to be normal, since the points are distributed around the diagonal line of the Q-Q Plot diagram shown in Figure 6 (a). In Figure 6 (c), in turn, the assumption of homoscedasticity can be confirmed, since the maximum dispersion of the points is constant around the line. Finally, the last assumption, independence among the errors, can be verified in Figure 6 (d), since there is no apparent linear pattern in the distribution of points.

So, in the diagrams of residual analysis, we could verify that all the assumptions – normality, homoscedasticity and independence of the errors – about the structures of the errors of the LRM selected for the Set C were met.

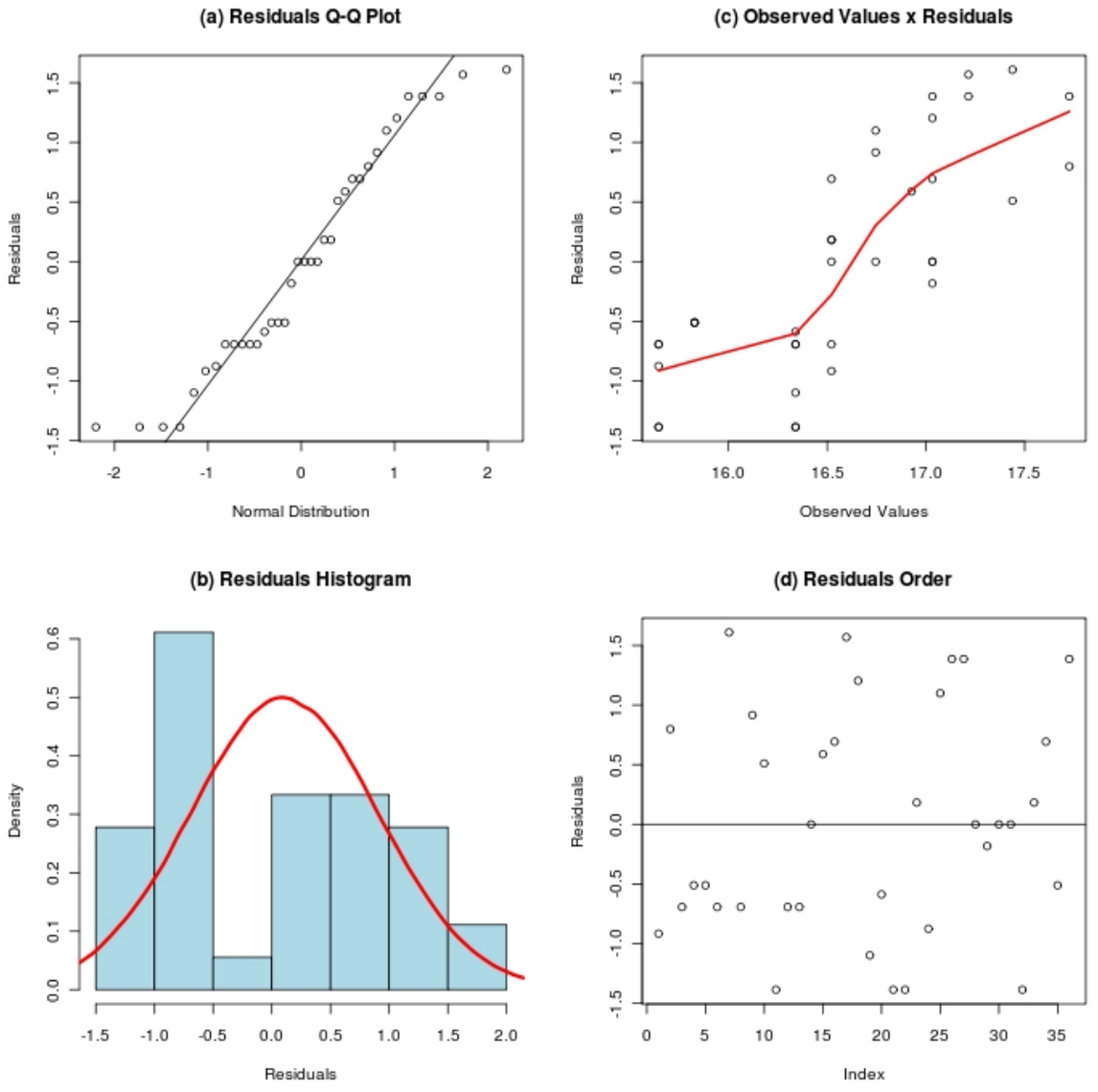


Figure 6. Graphics for analysis of assumptions about the distribution of errors for the training set with 50% of the project space.

Measurement	Set A	Set B	Set C
***Mann-Whitney-Wilcoxon* test (P-Value)**	100%	100%	79.12%
Global mean error	7.81e-08%	0%	7.15e-06%
Maximum error	1.43e-07%	0%	4.52e-05%
Minimum error	0%	0%	1.88e-08%

Table 5. Testing the fitness to the data from the training set and global mean, maximum and minimum errors for the LRMs selected for the Sets A, B and C.

In order to check the adherence of the LRMs to the data of the respective training sets, we performed the *Mann-Whitney-Wilcoxon* test, besides the computation of the global mean, maximum and minimum errors. The results can be seen in Table 5.

According to the result of the *Mann-Whitney-Wilcoxon* test, presented in Table 5, we can see that the estimates, given by the LRMs selected for the Sets A, B and C, tend to be equal to the data in the respective training sets, since the p-values are above the significance level, defined in the test as 5%. Analyzing Table 5, still, we notice that the selected LRMs presented accurate estimates, since the mean global, maximum and minimum errors were almost zero.

Still analyzing the precision of the estimates, with respect to the Set C, the diagram of accumulated errors is presented in Figure 7. It shows the cumulative error (x axis) for percentages of the training set (y axis). The accumulated errors indicate the deviation between the estimates given by the LRM and the data from the training set. In this case, the estimates given by the selected LRM differed by a maximum of 5e-07.

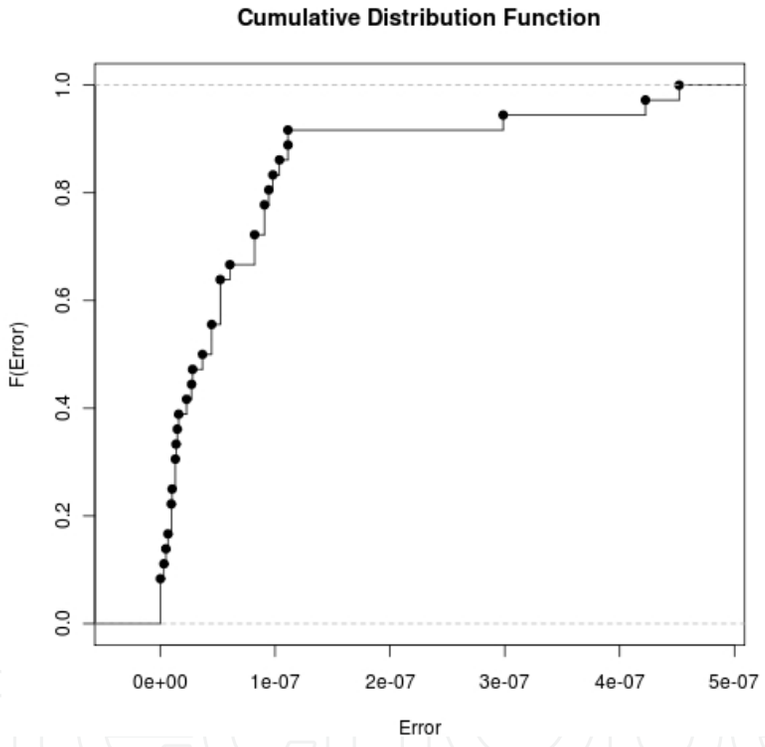


Figure 7. Graphic of accumulated errors for the LRM selected for the Set C.

Finally, in order to evaluate the precision of the predictions, which are the estimates given for the respective test sets of the Sets A, B and C, the selected LRMS were submitted to the *Mann-Whitney-Wilcoxon* test. Besides this test, the global mean, maximum and minimum errors were computed. The results can be seen in Table 6.

In Table 6, according to the results of the *Mann-Whitney-Wilcoxon* test, defined with a significance index of 5%, for the three sets, the estimates given by the selected LRMs tend to be equal to the data of the respective test sets. The three models had values for the global mean and minimum errors very close. For the maximum errors, there was a little variation, with the LRMs selected for the sets B and C, obtaining the highest and the lowest indexes, respectively.

Measurement	Results		
Set	**A**	**B**	**C**
***Mann-Whitney-Wilcoxon* test (P-Value)**	53.05%	69.11%	59.25%
Mean global error	4.12%	4.15%	4.75%
Maximum error	11.11%	14.21%	9.23%
Minimum error	4.905e-05%	9.171e-02%	8.27e-06%

Table 6. Test of fitness to the data of the test set and the global mean, maximum and minimum errors.

Still analyzing the results of the measurements presented in Table 6, we notice that the indexes obtained for the three sets, were comparatively very close. Such results may be explained by the used of the technique of selection of the training sets, which returns samples with high representative power.

In general, the use of the approach proposed in this work, which added methods for evaluation of the LRMs selected by the GP algorithm and the technique of selection of the elements of the training sets, allows the obtainment of solutions capable of providing precise estimates, even with the use of small samples.

6. Conclusions

This work has described an approach for obtainment and formal validation of LRMs, by means of the combination of genetic programming with statistical models. Our approach used the Audze-Eglais Uniform Latin Hypercube technique for the selection of samples with high representative power to form the training set. In order to evaluate the LRMs found with the introduced technique, we used statistical tests of hypothesis and residual analysis, aiming to verify the assumptions about the structures of the errors of these models.

In order to validate the proposed approach, we used a case study, with the prediction of performance in embedded systems. The problem of the case study consisted in exploring the configurations of a data bus in order to optimize the performance of the embedded application of sorting a set of integers by radix. So, with the use of the proposed technique, we generated LRMs capable of estimating the performance for all of the bus configurations.

The validation stages allowed us to realize that the LRMs found are adequate to the prediction of performance of the application, since all the assumptions about the structures of the errors were verified. So, the final LRMs were able to estimate the performances accurately, presenting mean global errors below 5%.

Author details

Guilherme Esmeraldo[1,3,*], Robson Feitosa[1], Dilza Esmeraldo[2], Edna Barros[3]
[1]*Federal Institute of Ceará, Crato,*
[2]*Catholic College of Cariri, Crato,*
[3]*Federal University of Pernambuco, Recife, Brazil*

[*] Corresponding Author

Acknowledgement

This paper has been supported by the Brazilian Research Council - CNPq under grant number 309089/2007-7.

7. References

[1] Augusto D.A (2000) Symbolic Regression Via Genetic Programming. In Proceedings of Sixth Brazilian Symposium on Neural Networks, Rio de Janeiro.
[2] Koza J.R (1992) Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press.
[3] Spector L, Goodman E, Wu A, Langdon W.B, Voigt H.M, Gen M, Sem S, Dorigo M, Pezeshk S, Garzon M, Burke E (2001) Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann.
[4] Keijzer M (2003) Improving Symbolic Regression with Interval Arithmetic and Linear Scaling.In Ryan C, Soule T, Keijzer M, Tsang E, Poli R., Costa E, editors. Heidelberg: Springer. 70-78 pp.
[5] Esmeraldo G, Barros E (2010) A Genetic Programming Based Approach for Efficiently Exploring Architectural Communication Design Space of MPSOCS. In Proceedings of VI Southern Programmable Logic Conference.
[6] Paterlini S, Minerva T (2010) Regression Model Selection Using Genetic Algorithms, Proceedings of the 11th WSEAS International Conference on RECENT Advances in Neural Networks, Fuzzy Systems & Evolutionary Computing.
[7] Wolberg J (2005) Data Analysis Using the Method of Least Squares: Extracting the Most Information from Experiments. Springer.
[8] Sakamoto Y, Ishiguro M, Kitagawa G (1986) Akaike Information Criterion Statistics. D. Reidel Publishing Company.
[9] Seber G. A. F, Lee A.J (2003) Linear Regression Analysis. Hoboken: Wiley.
[10] Weisberg S (2005) Applied Linear Regression, Third Edition. Hoboken: Wiley.
[11] McCulloch C.E, Searle S.R (2001) Generalized, Linear and Mixed Models. New York: Willey.
[12] Anderson D, Feldblum S, Modlin C, Schirmacher D, Schirmacher E, Thandi E (2004) A Practitioner's Guide to Generalized Linear Models. Watson Wyatt Worldwide.
[13] Hausmana J, Kuersteinerb G (2008) Difference in Difference Meets Generalized Least Squares: Higher Order Properties of Hypotheses Tests. In Journal of Econometrics, 144: 371-391.
[14] Nelder J.A, Wedderburn R.W (1972) Generalized linear models. Journal of the Royal Statistical Society Series A, 135 (3): 370–384.
[15] Chellapilla K (1997) Evolving Computer Programs Without Subtree Crossover. In IEEE. Transactions on Evolutionary Computation, 1(3):209–216.
[16] Aho A.V, Lam M.S, Sethi R, Ullman J.D (2006) Compilers: Principles, Techniques, and Tools, Second Edition. Prentice Hall.

[17] Antony J (2003) Design of Experiments for Engineers and Scientists. Butterworth-Heinemann.
[18] Cox D.E (2000) The Theory of the Design of Experiments. Chapman and Hall/CRC.
[19] Mitchell M (1999) An Introduction to Genetic Algorithms. MIT Press.
[20] Dean A, Voss D (1999) Design and Analysis of Experiements. Springer.
[21] Audze P, Eglais V (1977) A new approach to the planning out of experiments. Problems of dynamics and strength, volume 35, 1977.
[22] Bates J.S, Sienz J, Langley D.S (2003) Formulation of the Audze-Eglais Uniform Latin Hypercube Design of Experiments. Adv. Eng. Software, 34(8): 493-506.
[23] GPRSKit. Genetically Programmed Respone Surfaces Kit. Available: http://www.cs.berkeley.edu/~hcook/gprs.html. Accessed 2012 April 13.
[24] Koza J.R (1998) Genetic Programming On the Programming of Computers by Means of Natural Selection. MIT Press.
[25] Gen M, Cheng R (2000) Genetic algorithms and engineering optimization.Wiley.
[26] Ahn C.W, Ramakrishna R.S (2003) Elitism-Based Compact Genetic Algorithms. IEEE Transactions On Evolutionary Computation, 7(4).
[27] Koza J.R, Poli R (2005) Genetic Programming, In Edmund Burke and Graham Kendal, editors. Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. Springer.
[28] Sprent N, Smeeton N.C (2007) Applied Nonparametric Statistical Methods, Fourth Edition. Chapman and Hall/CRC.
[29] Fay M.P, Proschan M.A (2010) Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. Statistics Survey, 4: 1-39 pp.
[30] Shapiro S.S, Wilk M.B (1965) An analysis of variance test for normality (complete samples). Biometrika 52 (3-4): 591–611 pp.
[31] Breusch T.S, Pagan A.R (1979) Simple test for heteroscedasticity and random coefficient variation. Econometrica (The Econometric Society) 47 (5): 1287–1294 pp.
[32] Savin N.E, White K.J (1977) The Durbin-Watson Test for Serial Correlation with Extreme Sample Sizes or Many Regressors. Econometrica 45(8): 1989-1996 pp.
[33] Woo S.C, Ohara M, Torrie E, Singh J.P, Gupta A (1995) The SPLASH-2 Programs: Characterization and Methodological Considerations. In Proceedings of the 22nd International Symposium on Computer Architecture Santa Margherita: 24-36 pp.
[34] Cormen T.H, Leiserson C.E, Rivest R.L, Stein C (2001) Introduction to Algorithms. McGraw-Hill and The Mit Press.
[35] Black D.C, Donovan J (2004) SystemC: From the Groung Up. Kluwer Academic Publishers.
[36] ARM AMBA (1999) AMBA Specification rev. 2.0, IHI-0011A, May 1999. Available: http://www.arm.com/products/system-ip/amba/amba-open-specifications.php. Accessed 2012 April 13.
[37] Madar J, Abonyi J, Szeifert F (2005) Genetic Programming for the Identification of Nonlinear Input–Output Models. In Industrial and Engineering Chemistry Research, 44: 3178 – 3186 pp.