

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

4,800

Open access books available

122,000

International authors and editors

135M

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



Self Adaptive Genetic Algorithms for Automated Linear Modelling of Time Series

Pedro Flores¹, Larysa Burtseva² and Luis B. Morales³

¹*Universidad de Sonora*

²*Universidad Autónoma de Baja California*

³*Universidad Nacional Autónoma de México
México*

1. Introduction

In this work it is developed a methodological proposal to build linear models of Time Series (TS) from setting out the problem of obtaining a good linear model, such as solving a problem of nonlinear optimization with bounded variables. It is worth to mention that to build these problems are taken some ideas of the traditional statistical approach.

As product of the methodology here presented, it will be developed two heuristic algorithms for the treatment of TS, which allow building several models for the same problem, where the accuracy of these can be increased by increasing the number of terms of the model, situation that does not happen with the traditional statistical approach. Thus, with this algorithms it can be obtained several proposals of solution for the same problem, of which it can be selected the one that presents the best results in the forecasting. In addition, the algorithms proposed in this work allow building different linear versions, but equivalent to the Autoregressive (AR) and the classic Autoregressive with Moving Average (ARMS) models, with the added advantage of the possibility of obtaining models for not stationary TS, and with non stationary variance, in cases where the traditional methodology does not work.

Since optimization problems set out here may present multiple local minimums, it is needed to use a special technique to solve them. With this end it was developed a version of the Self Adaptive Genetic Algorithms (SAGA), encoded on real numbers that allows, without intervention of the user, to find satisfactory solutions for different problems without making changes in the parameters of the code.

On the other hand, among the principal points of this methodology it is the fact that in many cases, these linear versions present a phenomenon that has been named 'forecasting delay', which allows to modify the linear model obtained to find a more accurate forecasting.

It is important to notice that the first AR version of the algorithms developed for the TS were tested in the examples of the international competition:

"NN3 Artificial Networks & Computational Intelligence Forecasting Competition"

that from now on it will be called NN3, which was realized in 2006-2007 (<http://www.neural-forecasting-competition.com/NN3/results.htm>). This competition is

held annually to evaluate the accuracy of methods in the area of Computational Intelligence in diverse problems of TS. In this edition the problem at hand was to forecast with the same methodology 18 future values of a set of series where the majority are measurements of real phenomena. The competition has two categories: the NN3-Complete has 111 problems and the NN3-Reduced consists of 11 problems. In this competition using only models with four terms it was obtained the third place in the category NN3-Complete from 29 competitors, and the sixth in the category NN3-Reduced from 53 competitors. This work will be referenced in various sections in relation to the examples of this competition. An analysis of the results of NN3 can be found in (Crone & Hibon & Nikolopoulos, 2011)

2. Methodology

The forecasting process consists in calculating or predicting the value of some event that is going to happen in the future. To realize adequately this process it is needed to analyze the event data in question and build a model that allows the incorporation of the behavior patterns that have occurred in the past under the assumption that they can happen again in the future. It is important to note that there is not interest in explaining how the mechanism that produces the events works, but to predict their behavior.

The TS models are used for studying the behavior of data that varies with time. The data can be interpreted as measurements of some value (observable variable) of a phenomenon, realized at time intervals equal and consecutive. There are several methods to construct TS models and an overview of the most important can be found in (Weigend & Gershenfeld, 1994). In (Palit & Popovic, 2005) it is shown an overview of the methodologies most used in the area of computational intelligence. One of the most used methods is based on considering the TS as a realization of a stochastic process. This approach is the basis of statistical treatment of TS that can be found in (Box & Jenkins, 1976) and (Guerrero, 2003). Nowadays the construction of model for TS is an area of great development as evidenced by the articles of the Journal of Time Series Analysis (<http://www.wiley.com/bw/journal.asp?ref=0143-9782&site=1>) in addition to the papers presented in international competitions on time series modelling such as NN3. Nevertheless the existence of GA papers in which are used the TS (Alberto et al, 2010; Battaglia & Protopapas, 2011; Chiogna & Gaetan & Masarotto, 2008; Hansen et al, 1999; Mateo & Sovilj & Gadea, 2010; Szpiro, 1997; Yadavalli et al, 1999), it is important to note that it was not found any reference to the use of SAGA for this purpose.

The data will be represented by $\{Z_t\}$ with the implicit assumption that t takes the values $1, 2, \dots, N$ where the parameter N indicates up to what moment the information is had. When it is had a model for the data set, then it can be estimated values for the TS, which are denoted by $\{F_t\}$. In addition in order to consider a model as a good one, it is required that the values of $\{F_t\}$ be "similar" to those of $\{Z_t\}$. The main purpose of this work is to build linear models for the data set to have good estimates of the K unknown values of the phenomenon being studied in the moments $N + 1, N + 2, \dots, N + K$.

In the forecasting subject, when it is had a TS with these $N + K$ data, the set of the first N is called training set, and is used to construct the model of the series and realize the estimation of its parameters. The set of the last K terms in the series is called training set, and is used for the comparison of different models to choose the most suitable. Especially, it is been interested in building automated Autoregressive models of order p (AR (p)). For the TS are expressions

of the form:

$$Z_t = \delta + \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t \quad (1)$$

Where Z_t is the observable variable in question, δ and ϕ_i are the parameters to be determined and the variable a_t represents a random variable of noise called *residual*. The expression (1) means that to predict what will happen at the time t are required the p values previous to t , these values are called *delays* or *lags*.

In the classic theory of linear models is set the restriction that a_t represents a white noise, but in this work it was not included this boundary, which will allow to find AR expressions for the residuals with which it will be possible to increase the accuracy of the models.

The interest in this type of models is originated in the fact that they represent the most important information about the behavior of the series eliminating the noise that may appear. It should also be added that, for these models, it is important that in the expression (1) only appears a number of terms set in advance. This will allow finding models for a TS, controlling the accuracy of the approximation of the same series according to the number of terms utilized.

Problem 1: If $\{Z_t\}$ is the original TS and $\{F_t\}$ is the forecasting obtained of the form

$$F_t = \delta + \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} \quad (2)$$

with $t > p$.

It is necessary to find the values for δ and ϕ_i that minimize the function:

$$\sqrt{\sum_{i=p}^N (Z_i - F_i)^2} \quad (3)$$

This function will be called *Root of the Sum of Squares* (RSS). It is necessary to add that for rapidity in calculation it is preferable to use the square of this function obtaining the same results.

In this initial setting out the construction of the model is presented as if a linear interpolation problem was solved, and given that the values for δ and ϕ_i will not be arbitrary but will be looked at certain intervals are necessary methods to solve the Problem 1 working in addition with bounded variables.

The RSS function can have multiple local optima, and to solve this problem it was developed an original version of SAGA algorithms, which allows to solve real nonlinear optimization problems and with bounded variables. The selection of a self Adaptive version was carried out by the fact that it is wanted to automate as much as possible the process of building these models.

3. Self adaptive genetic algorithms

The SAGA algorithms were developed by Thomas Bäck (Bäck, 1992a, 1992b) and have the characteristic that they alone look for the best parameters for their operation. In them, the parameters that will be self Adaptive are encoded in the representation of the individual, for which they are altered by the actions of the genetic operators. With this, the best

values of these parameters will produce better individuals, which have major probability of surviving, and in consequence, will spread towards the whole population the best values of the parameters. There are several versions of SAGA that differ especially in the parameters that will be adjusted automatically (Eiben et al., 1999). In the case of this work four self Adaptive parameters are used: *individual probability of crossing* p_c , *repetition of crossing* r_c , *individual probability of mutation* p_m and *repetition of mutation* r_m as presented in (4). The selection of these parameters and its values is based on the idea that genetic operations of crossing and mutation can be multiple, but they cannot have very large values. A binary version of this algorithm already had been used by one of the authors of this work in other problems (Flores, 1999; Garduño, 2000; Garduño, 2001; Sanchez, 2004), and this, as well as the presented one here (with the representation of real numbers) according to the literature reviewed, are original of himself.

The individuals for these problems will be proposed as solutions to them, and in addition will have four more components, where it will be represented the values of: *individual probability of crossing* p_c , *repetition of crossing* r_c , *individual probability of mutation* p_m and *repetition of mutation* r_m . To this section of the individual it is called section of the self Adaptive parameters, and with this, our entire individual is represented by:

$$(\delta, \phi_1, \phi_2, \dots, \phi_p, p_c, r_c, p_m, r_m) \quad (4)$$

The above mentioned is necessary, so in this model, the probability of crossing and mutation will be characteristic of each individual (not of the population as is traditional in the GA), and in addition it is considered that the crossing and the mutation can be multiple, that is to say, to operate several times in the same time. The multiple crossing and mutation are repetitions of the crossing and mutation that are used in the GA, when are used individuals represented by vectors of real components. The way of operating with these parameters is similar to that presented in (Bäck, 1992a, 1992b).

The limits that were used in the code of this work for the self Adaptive parameters are: *individual probability of crossing* p_c that changes in the interval (0.5, 0.95), *repetition of crossing* r_c in (1.0, 4.0) what means that only can be crossed from one to three times, *individual probability of mutation* p_m that varies in (0.5, 0.85) and *repetition of mutation* r_m in (1.0, 5.0) what means that just it is possible to mutated from one to four times. The limits of these self Adaptive parameters were chosen on the basis of the experience of other works (Flores, 1999; Garduño, 2000; Garduño, 2001; Sanchez, 2004), where they proved to give good results.

Later there are detailed the procedures of crossing and mutation.

3.1 Crossing and mutation

Given two individual, the crossing is realized taking as probability of crossing the average of the values of the individual crossings. Once it has been decided if the individuals cross, it is taken the integer part of the average individual crossing, and that is the number of times they cross. The crossing of two individuals consists of exchanging the coordinates of both vectors from a certain coordinate chosen at random. The multiple crossing is the result of applying this procedure several times to the same vectors.

For the mutation it is taken the individual probability of mutation of the individual, and accordingly to this it is decided whether mutated or not. As soon as has been decided that

an individual mutates, this is mutated as many times as the value of the integer part that has in the repetition of mutation of himself. To apply the mutation to an individual a coordinate of the vector is chosen at random, and it is changed its value for another (chosen also at random) between the limits established for the above mentioned coordinate. The multiple mutation is the application of this procedure to the same individual several times.

3.2 Use of the self adaptive genetic algorithms

Since SAGA are random, it is common to realize several runs on the same problem and choose the best result from them. These algorithms are applied in three stages to solve our problems. In the first two stages are defined which are the important variables to solve the problem, and in the third stage, it is where the solution is calculated properly. It is important to note that the individuals have two parts, but in this section only there is born in mind the first part of the individual, which corresponds to the Autoregressive components. Here is the procedure based on SAGA, which is performed to obtain a solution to the problem.

In the first stage are used SAGA to explore the space of solutions and later to define which variables among $\delta, \phi_1, \phi_2, \dots, \phi_p$, are the most important for the problem in question. For this were done 10 repetitions of 1000 iterations each, and with the solutions of each repetition, a vector is constructed by the sum of the absolute values of $\delta, \phi_1, \phi_2, \dots, \phi_p$. (see Figure 1)



Fig. 1. Solution using all variables.

In this first stage, the aim is to realize an exploration of the space of solutions, and for that are performed 10 iterations with all variables to consider. Then, with the 10 solutions obtained, a vector is built by adding the 10 solutions with all its positive components, and it is assumed that the largest values of these components are the most important.

In the second stage the SAGA are applied to find solutions by considering only the important variables of the problem. For this is defined in advance how many variables are required (this will be seen to detail below), and are chosen those which correspond to larger values of the first stage. In this stage 5 repetitions are realized, where each one is finished until the optimum is not modified in the last 200 iterations. Of these 5 repetitions the best result obtained is chosen (see Figure 2).

In this second stage, only are considered the variables that had greatest values in the part of the autoregressive components of the individual, and for them are kept the original intervals of its values: For all the other variables in this part of the individual, it is stated that the upper and lower limits are zero. In this stage 5 repetitions are realized and from them is chosen the one that has lower value of RSS.

In the third stage it will be found the solution in which only are taken into account the important variables obtained in the previous stage. For this are extended the boundaries

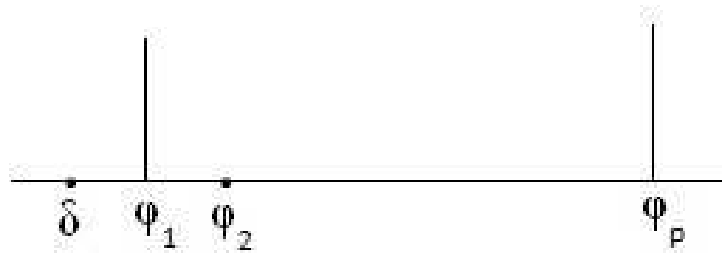


Fig. 2. Solution using the most important values.

of the variables of the solution obtained in the previous stage, which absolute value is greater than 0.01. The upper limits of the variables considered are the nonzero values obtained in the previous best solution of more than 1.0 and the lower with less than 1.0. The upper and lower limits of the other variables of the autoregressive components of the individual will be zero. With these limits is solved once until the optimum is not improved in 250 iterations. Since the GA are random for each problem were performed 5 iterations, and of them it was chosen the best.

The main characteristics of the SAGA version used in this work that make them original are:

- Real coding is used for the variables of the problem. This allows a more simple code that can easily pass from one stage to another of those presented here.
- The probabilities of crossing and mutation are characteristics of each individual and the crossing and mutation procedures are established on the basis of these individual characteristics.
- The repetitions of crossing and mutation are multiple though the values that take are not very big.
- It was introduced a control mechanism that prevents the proliferation within the population of the best individual copies, thus eliminating the risk of premature convergence.

Above all, the last three features are inspired by the fact that the nature behavior is more flexible than rigid, and therefore should be allowed more variability within the SAGA.

The main disadvantage that has the use of SAGA, is the major computational cost compared with traditional versions, but the advantage that is obtained is that with the same code it is possible to solve automatically all the problems of nonlinear modelling of TS.

4. Autoregressive models

The TS linear models are important because there are many applications where linear estimations are sufficient, besides they have a wide use in industrial situations. On the other hand, are also important because there are other methodologies that use forecasting (Medeiros & Veiga, 2000,2005). The classic reference for the treatment of linear models is (Box & Jenkins, 1976).

In the specific case of the AR that we care for TS, the value at a certain time should be calculated as a linear expression of the values of a certain number of previous measurements, as described in (Box & Jenkins, 1976). The AR models developed here fulfill the stochastic process of the residuals $\{a_t\}$ associated with them; it is not a white noise. The latter will allow

that once it is built a good AR model of TS, it can be build for it another AR model for residuals $\{a_t\}$, which together with the original one allow obtaining the equivalent of an ARMA model, but with major forecasting possibilities.

On the other hand, to solve the problem 1 it is first necessary to address the following questions, taking into account that is necessary to find an AR model with K terms, where K is established beforehand:

1. How many p terms must be considered?
2. At what intervals are the coefficients of the linear expression?
3. What K terms are most appropriate to solve this problem?
4. What are the values of this K terms that minimize the function (3)?

The following summarizes the results of the BJ methodology that is used in our proposal.

4.1 Main results of the Box Jenkins methodology

Univariate TS were analyzed by the Box-Jenkins (BJ) methodology from the formulation of equations in differences with a random additive component denominated white noise. For these BJ models the conditions in which is presented the stationarity property of the series and the scheme that has to be follow to determine the parameters of the particular model were studied.

The most general model is denominated ARMA(p,q) (Autoregressive Moving Average) and indicates the presence of autoregressive components both in the observable variable $\{Z_t\}$ as well as in the white noise $\{a_t\}$. A particular class of model for stationary series corresponds to the Autoregressive models AR(p) (that are denoted as AR), which is represented by the expression:

$$Z_t = \delta + \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t \quad (5)$$

When the series is stationary δ and ϕ_i are constants that satisfy the following relations:

$$|\phi_1| < 1, \quad (6)$$

$$\mu = \frac{\delta}{1 - \sum \phi_i}$$

and

$$\sum \phi_i < 1$$

Where μ represents the average of the series $\{F_t\}$. The relations in (6) are a consequence of the stationarity property and can be consulted in (Box & Jenkins, 1976).

The correlation structure presented by a TS related to an AR model for separate observations k time units is given by the autocorrelation function:

$$p_k = \phi_1 p_{k-1} + \phi_2 p_{k-2} + \dots + \phi_p p_{k-p}$$

where p_k is the autocorrelation for data of series separated k time units. From the initial conditions that satisfy this equation in differences are presented the following possible

behaviors: exponential or sinusoidal decay. This permits to determine if a series is stationary or not.

The most general model is the model ARIMA (p, d, q) (AutoRegressive Integrated Moving Average processes) that includes not stationary series for which apply differences of order d to stationarize it:

$$\phi_p(B)\nabla^d z_t = \delta + \theta_q(B)a_t$$

Where $\phi_p(B)$, $\theta_q(B)$ and B are operators that satisfy the following relations:

$$\phi_p(B)z_t = (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) = z_t - \phi_1 z_{t-1} - \dots - \phi_p z_{t-p}$$

and

$$\theta_q(B)a_t = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)a_t = a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$

$$B^k z_t = z_{t-k}$$

$$\nabla^d z_t = (1 - B)^d z_t$$

Similarly, there is a general model that considers the presence of stationarity or cyclic movement of short term of longitude s modeled by the expression:

$$\phi_P(B^s)\phi_p(B)\nabla^d z_t = \delta + \theta_Q(B^s)\theta_q(B)a_t$$

Where $\phi_P(B^s)$ y $\theta_Q(B^s)$ are polynomial operators similar to the above mention, but its powers are multiples of s, $\{a_t\}$ are residuals in the moment t and θ_t are its components in the part of moving averages.

BJ methodology satisfies the following stages:

- (a). Identification of a possible model among the ARIMA type models. To accomplish this first is necessary to determine if the series is stationary or not. When an observed series is not stationary the difference operator is applied:

$$\nabla z_t = z_t - z_{t-1}$$

as many times as it will be necessary up to stationarity. To avoid overdifferentiation it is calculated the variances of the new obtained series choosing the one with the smallest value.

When a series is stationary in its mean, but its variance is increasing or decreasing according to BJ methodology it should be applied a transformation (generally logarithmic) for the stability of the variance. It is important to notice that this is not necessary in our proposal.

Given a stationary series the behavior pattern of the autocorrelation function and the partial autocorrelation indicate the possible number of parameters i and j that the model should have.

Besides the presence of stationarity in a temporal series there is other property that is required in the ARIMA models denominated invertibility, which permits to represent the series as an autoregressive model of infinite extension that satisfy the condition:

$$\lim_{i \rightarrow \infty} \phi_i = 0$$

The above mention allows that with a finite number of terms could be obtained an expression that satisfies the form (1) for the series. This means that only the ARIMA models that have the invertibility property can be approximated by an AR model of the form (1).

- (b). Estimation of the model parameters by means of non linear estimation techniques.
- (c). Checking that the model provides an adequate fitting and that the basic assumptions implied in the model are satisfied through the analysis of the residuals behavior. Is important to mention that our proposal does not need such analysis because the residuals do not correspond, in general, to the white noise.
- (d). Use of the model.

Next are presented the characteristics of the heuristic proposed algorithms. Note that these algorithms are used to build models AR of TS since the ARMA models are built from these.

4.2 Proposed algorithms

The heuristic algorithms built in this work are based in the following assumptions:

- (a). Regardless the original series type (stationary or non stationary) the model looked will always be of the form AR presented in (1).
- (b). To determine how many delays p are required, first is necessary to choose the differences series that will be used to estimate these, afterwards it is defined the number of delays according to the behavior of the autocorrelation sample function of the difference series chosen. This implies a difference with the BJ methodology, which applies the number of delays under the terms of the information that provides both the autocorrelation function as well as the partial autocorrelation function and the hypothesis of the random component as white noise. This choice has as consequence in the models developed here that at will not be white noise.
- (c). The conditions of (6) become more relax, since in spite of be satisfied it in the stationary series, in this work these will be applied to series that could not be stationary.

It is necessary to add that the heuristic algorithms presented here allow the treatment of series with trend and variance time-dependant, since they do not require the conditions that traditionally are asked to the TS, as is the fact that they are stationary or of stationary variance or that they result from applying a logarithmic transformation or moving averages.

The first algorithm that we propose builds a linear approximation for the series of differences (of first, second or third order) that could be stationary. Then, from this linear approximation and using the result 1, it is built another linear model of the original series.

4.2.1 First algorithm

In this stage, first it is decided which series will be used to work with among the original, the first differences, the second differences and in our case it is included the possibility of working with third differences series. In order to decide this it is chosen the series that have the lowest variance, which we consider as an indication of having a stationary series (Box & Jenkins, 1976).

Once that was chosen the series to work with it will be estimated how many terms are necessary for the linear approximation of the series with base in the autocorrelation function. In this work were calculated 30 values for the autocorrelation function and for selecting how many terms are required two cases were utilized. If the function is decreasing a value of 4 is taken, on the contrary a value equal to the value in which the first maximum of this function is observed it will be chosen (see Figure 3). With this procedure if the series presents stationarity and the period is smaller than 30 the models that are built here can represent appropriately such stationarity.

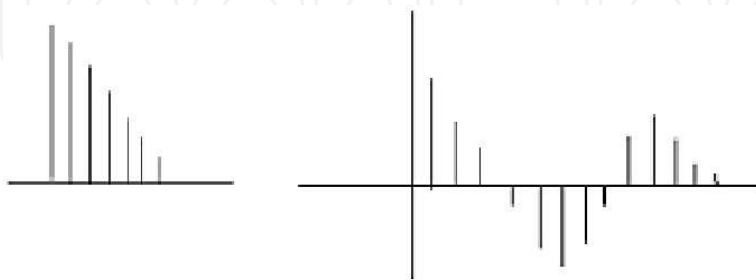


Fig. 3. Possible autocorrelation function graphs.

With this information are built the limits for the coefficients intervals of the chosen series, for that are taken all the ϕ_i in $[-1, 1]$ except the independent term δ which limits are calculated between zero and the average value of the series. The reason why these limits are established is obtained from the equations presented in (6) With all the previous information it is complete the proposal of the p number of terms required and that are the limits of its coefficients. From this information is solved the **problem 1** applying the SAGA in the first two stages depicted in section 3.2 with base on the following:

Result 1. If $\{y_t\}$ is a difference series for $\{x_t\}$ with a model

$$y_t = h_0 + h_1 y_{t-1} + h_2 y_{t-2} + \dots + h_k y_{t-k}$$

then, for the difference series with terms $y_t = x_t - x_{t-1}$ must be

$$x_t = h_0 + (1 + h_1)x_{t-1} + (h_2 - h_1)x_{t-2} + \dots + (h_k - h_{k-1})y_{t-k} - h_k y_{t-k-1} \quad (7)$$

is a model for the series $\{x_t\}$.

From this result two important consequences are obtained:

- The model for the series $\{x_t\}$ has one term more than the series $\{y_t\}$
- If y_t has a coefficient value between -1.0 and 1.0 , the coefficient of x_t may not be in this range.

Applying the **result 1** as many times as necessary, it can be obtained a model for the original series, and to this model it is applied the stage three of section 3.2 to obtain a linear model for the TS. Note that if it is had a model AR for some series of differences, the model built for the original series has more terms than the series of differences, so if K terms are needed for the original series, then must be found models for the series of differences of less terms that K .

4.2.2 Second algorithm

The second algorithm only utilizes of the BJ methodology the estimation of how many terms are necessary in the linear approximation of the series of differences, which could be stationary, thus, from this is determined the numbers of terms that will be used in the original series.

From now on, are applied the stages presented in section 3.2, taking the limits of all the coefficients in $[-1, 1]$, but always working with the original series. There is not a result that justifies the use of these limits, and only it has been found a reference (Cortez at all, 2004) where it is used. On the other hand is a fact that a high percentage of cases in the NN3 presented better results with this algorithm than with the first. As an example of this the second algorithm outperformed the first in 46 of the 111 examples of NN3-Complete.

5. NN3 results

The international competition *NN3 Artificial Neural Network & Computational Intelligence Forecasting Competition 2007* aims at assessing the latest methodologies for the forecasting of TS. This competition is open to use methods based on Neural Networks, Fuzzy Logic, Genetic Algorithms and others in the area of artificial intelligence. The problems in question are presented in two groups called NN3-Complete (with 111 examples of TS) and NN3-Reduced (with 11 examples), and the purpose of the competition is to obtain the best models for each example of the two sets using the same methodology. The notation of this section is similar to that used in NN3.

To evaluate the performance of a model in some example s , it is estimated the forecasting F and it is measured the performance with the average of the indicator *Symmetric Mean Absolute Percent Error* SMAPE in all the values of the series. The SMAPE measures the absolute symmetric error in percentage between the real values of the original series Z and the forecasting F for all observations t of the test set of size n for each series s with SMAPE equal to:

$$\frac{1}{n} \sum_{t=1}^n \frac{|z_t - f_t|}{(z_t + f_t)/2} * 100 \quad (8)$$

and finally it is averaged over all examples in the same set of data. Other measures of forecasting accuracy of a model can be found in (Hyndman & Koehler, 2006).

This indicator can evaluate the performance of applying different methodologies on the same set of data and the methodology that produces the lowest value is considered the best. In the set NN3-Complete the best result was of 14.84% and applying the algorithms developed in this work was of 16.31%. In the NN3-Reduced the results were 13.07% and 15.00% respectively. However, it is possible to build linear models with the methodology presented in this work to improve these results because:

- Although the competition was intended to determine the best model for each example in this work was found an AR model with 4 terms for each example. It is expected that if it is divided the series in a training set and in other set of test it can be found models with higher forecasting capacity that improve the results obtained.
- It were not used ARMA models that include the behavior of the residuals or the advancement of forecasting that substantially improve the results.

To build the NN3 competition models were conducted several activities. First it was worked the NN3-Reduced problems where, with the two algorithms developed, were realized 50 runs of every algorithm in each example looking for linear models with 4 terms. Table 1 presents the results of linear expressions and calculation of RSS.

After reviewing the behavior of the 50 solutions of these examples it was concluded that five runs were enough to obtain satisfactory results. For this reason only five runs were realized for the examples of the NN3-Complete using each algorithm and it was chosen the best of these. The results of the NN3-Complete examples are not presented.

Problem	Linear Model	RSS
101	$F_t = 1269.3358 + 0.3467Z_{t-1} + 0.6978Z_{t-12} - 0.2921Z_{t-13}$	1713.55
102	$F_t = 1.9987 + 0.9218Z_{t-1} + 0.9574Z_{t-12} - 0.8792Z_{t-13}$	5440.262
103	$F_t = 1.9989 + 0.9218Z_{t-1} + 0.8124Z_{t-12} - 0.3734Z_{t-13}$	80019.738
104	$F_t = 9.113 + 0.7252Z_{t-1} + 0.8316Z_{t-12} - 0.5592Z_{t-13}$	7321.538
105	$F_t = 1.998 + 0.9099Z_{t-1} + 0.3104Z_{t-11} - 0.2225Z_{t-13}$	1513.984
106	$F_t = 2821.9541 + 0.2673Z_{t-2} - 0.1699Z_{t-7} + 0.3422Z_{t-12}$	4464.87
107	$F_t = 0.9978 + 0.7937Z_{t-1} + 0.3152Z_{t-12} - 0.1125Z_{t-13}$	1387.011
108	$F_t = 2000.5819 + 0.2885Z_{t-2} - 0.1456Z_{t-4} + 0.2379Z_{t-5}$	10417.433
109	$F_t = 1.9988 + 0.9951Z_{t-1}$	2297.306
110	$F_t = 1863.0699 + 0.2520Z_{t-1} - 0.1058Z_{t-5} + 0.2359Z_{t-11}$	18593.279
111	$F_t = 474.1106 + 0.2420Z_{t-11} - 0.3319Z_{t-12} + 0.2688Z_{t-13}$	7248.281

Table 1. Linear models for the NN3-REDUCED.

5.1 NN3 graphs

In this section are showed some of the graphs of the series obtained with the best result of some heuristic algorithms here presented. The values correspondent to the last 18 points on the graph are the result of the forecasting obtained on having evaluated the expressions of the linear models that appear in Table 1.

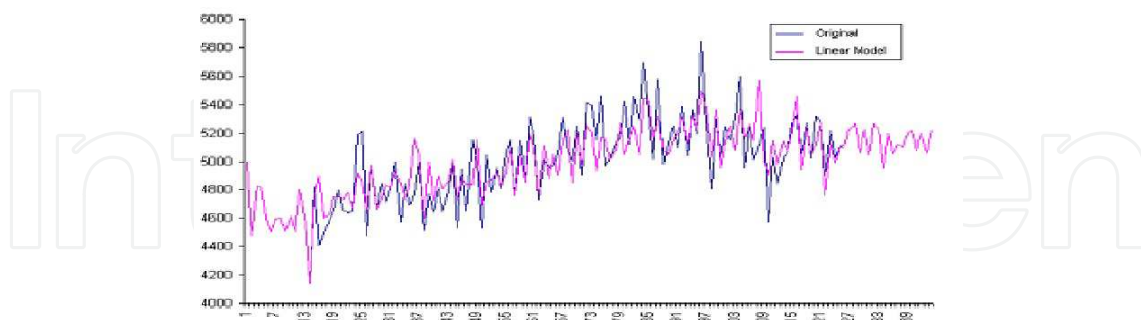


Fig. 4. Example 101.

6. ARMA models for time series

In this section the methodology already developed is applied to obtain AR components of the error series obtained by subtracting from the original series the values that are assigned by the AR model. With this is obtained a new model by adding these two components, thus it is obtained the equivalent in our methodology of the traditional ARMA models.

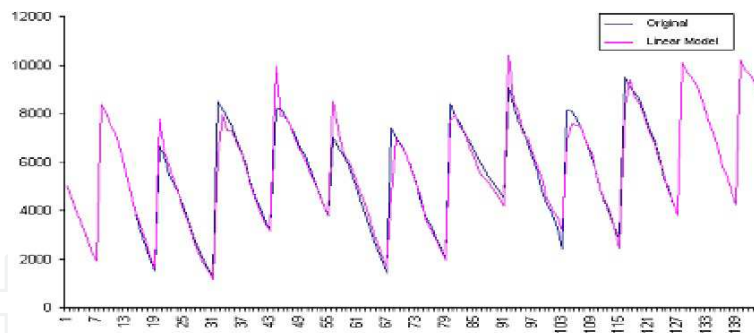


Fig. 5. Example 102.

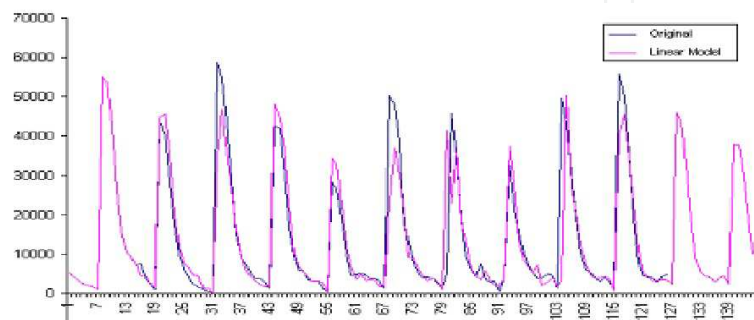


Fig. 6. Example 103.

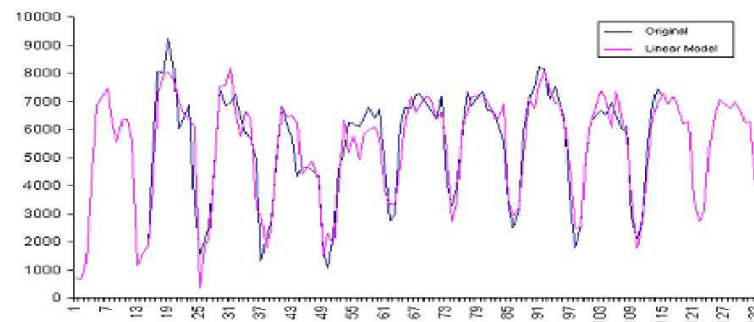


Fig. 7. Example 104.

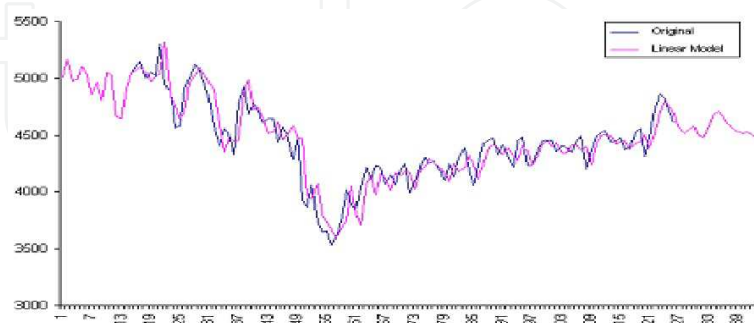


Fig. 8. Example 105.

In the first part of this section is presented, as an example, the Fig. 10 of the error obtained with our methodology for a certain series for a particular series that for its behavior it can be concluded that is not a white noise. Note that when are realized tests of white noise to the errors obtained with this methodology it was not observed that this was a white noise.

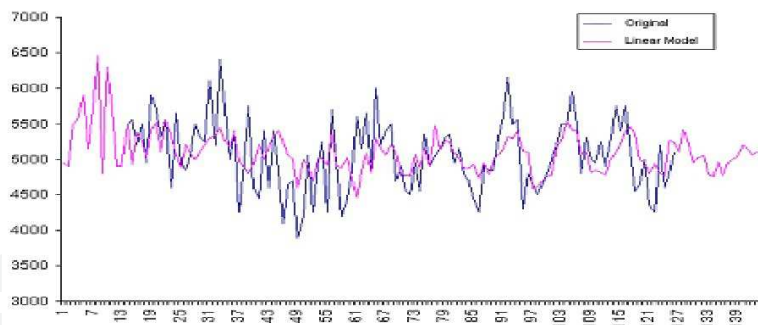


Fig. 9. Example 106.

Therefore it can be built AR models for these error series, which will have the capability to adequately model the error, which allows, when considering these two models, to obtain a bigger forecasting capability.

6.1 Building of the ARMA models

The most general models used in this work are the Autoregressive Moving Averages ARMA (p, q) that contain the presence of autoregressive components in the observable variable Z_t and in the error a_t , where:

$$a_t = Z_t - \left(\delta + \sum_{i=1}^p \phi_i Z_{t-i} \right)$$

and

$$F_t = \delta + \sum_{i=1}^p \phi_i Z_{t-i} + \sum_{j=1}^q \gamma_j a_{t-j}$$

Once the AR model is obtained for a series it can be built an ARMA model from the acquiring other AR model for the series obtained when considering the a_t errors between the original series and its AR model. When is added to the AR model an additional component that considers the autoregressive terms corresponding to the error is obtained the complete ARMA model. Figure. 10 shows an example of the error for the series.

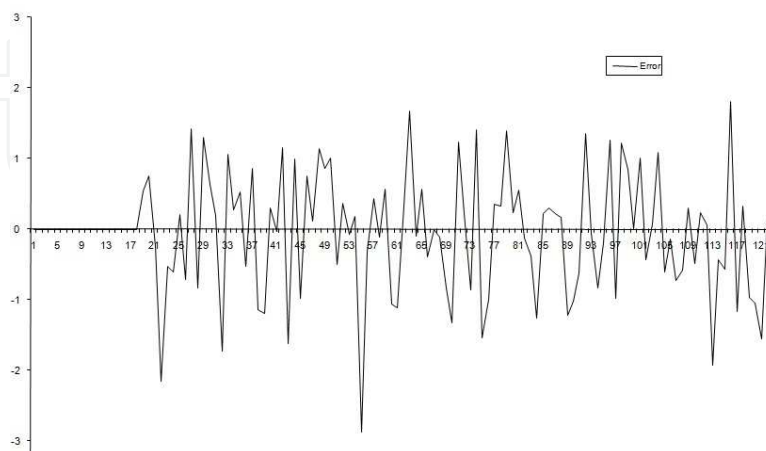


Fig. 10. Example of a TS corresponding to the error.

The procedure to build the ARMA models is realized in two stages. First is built an AR model for the original series, afterwards it is considered the error series a_t to which it is found other

AR model. In both procedures the most important stage is to define how many terms are required for each model.

From now on the ARMA notation for a series changes, for this it will be indicated to which part of the expression of AR or MA corresponds, and the constants ϕ_i and γ_j will represent the terms of the corresponding expression, in other words the terms F_{t-i} and a_{t-j} it will not be written.

7. The forecasting delay phenomenon

Analyzing the graphs of the built models with this methodology for the examples of the NN3-complete it was detected a phenomenon that visually appears as if the graph of the model were almost the same that the original series, but with a displacement of one unit to the right. This phenomenon was observed in the NN3-Complete in 20 examples: 51, 64, 66, 74, 80, 82, 83, 84, 85, 86, 88, 89, 90, 91, 92, 95, 100, 105, 107 and 109.

Given that the first 50 examples of the competition corresponded to series of 50 values (apparently built by experts) and the last 61 examples were series of 150 terms (seemingly of real phenomenon) it was supposed that the 34% of the real examples of the NN3 present this behavior. From this information we can assume that this phenomenon appears in a large percentage of the models built with this methodology and, for this reason the model built with this methodology will give better results when applying to these series. Following is showed in Fig. 11 an example of this phenomenon corresponding to the AR model of the example 74 obtained with the methodology of this work.

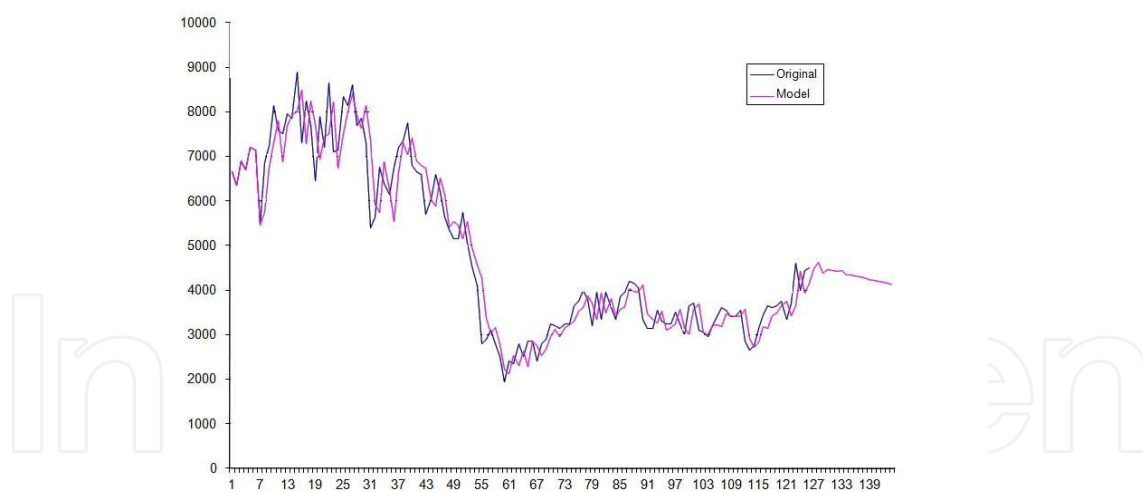


Fig. 11. Example 74 of the NN3-Complete.

This phenomenon was called in this work as *forecasting delay* (FD), since is equivalent to forecast in a certain moment what happen in the previous moment.

8. The procedure of advancement of forecasting

The FD phenomenon can be used by modifying the graph of the linear models obtained by applying a displacement of one unit to the left of its graph. This procedure was defined as *advancement of forecasting* (AF) and it is formalized next.

Definition: Be a time series with model AR or ARMA

$$F_t = \delta + \sum_{i=1}^p \phi_i Z_{t-i} + \sum_{j=1}^q \gamma_j a_{t-j}$$

The advancement of the forecasting was denominated as the following operation:

$$F_t = F_{t-1}, \text{ for } t > \max(p, q) \quad (9)$$

When is applied to an AR or ARMA this operation it is said that is a linear model AR or linear ARMA with AF respectively. In figure 12 is shown the linear model of the example 74 with AF.



Fig. 12. Example 74 of NN3-Complete to which it was applied the advancement of forecasting.

A first result obtained is that if a series that presents FD it is applied the AF, then the value of RSS for these models is smaller than the error of the original ARMA models. This is caused because when is displaced the graph of a model one unit to the left, which is what means the operation (9), almost it is superimposed to the graph of the original series. Extrapolating this behavior to the region of forecasting it is expected that the same effect occurs and that the values of the linear model with AF be a better approximation than those of the linear models. Due to the above it is supposed that the linear models with AF will have a better forecasting capacity. As an example, in Table 2 is showed the improvement of the linear models with AF for 10 examples of NN3 that present DF.

The improvement (imp) in the models here presented ranges from 10.28% to 97.27% with an average of 48.48%, and it is expected that as the percentage is greater the ability of the forecasting model increases by a similar proportion. It should be noted that when it is had an AR model with four terms it is very difficult to improve substantially the value of RSS by incrementing the terms of the AR model or including terms of the part of the moving averages.

Example	AR	RSS	RSS AF	imp %
51	$0.9968 + 0.8129_1 + 0.6594_{12} - 0.4799_{13}$	3767.009	3379.4502	10.28
64	$0.3894 + 0.9199_1 + 0.6778_{12} - 0.5999_{13}$	3899.114	3495.0649	10.36
66	$0.9984 + 0.9202_1 + 0.5858_{12} - 0.5103_{13}$	3893.0544	2803.1406	27.99
74	$0.9993 + 0.9448_1 + 0.5226_{12} - 0.4800_{13}$	4894.1655	3340.5911	31.74
86	$-0.9991 + 0.6235_1 + 0.2161_2 + 0.1907_{17}$	4114.1499	1917.1523	53.40
88	$0.9979 + 0.7001_1 + 0.1531_{11} + 0.1438_{18}$	2449.5383	1265.7606	48.32
89	$0.9995 + 0.8914_1 + 0.2169_{12} - 0.1079_{13}$	1247.8290	339.8757	97.27
105	$1.9980 + 0.9099_1 + 0.3104_{11} - 0.2225_{13}$	1513.984	664.8109	56.08

Table 2. Comparison of RSS for linear and linear with AF models.

9. Comparisons with other methodologies

To evaluate the performance of a model on a TS data is divided into two sets called training set and test set. The training set has the first values of the series (approximately 90% of the total) and the test set the last 10%. The information of the training set model is used to choose the model and evaluate the parameters. Once chosen the corresponding model is evaluated its ability to forecast the test set, and when it is had different model proposals it is common to choose the best result of the test set. For this assessment can be used several measures of performance (Hyndman & Koehler, 2006). In this work preferably is used RSS.

To build the models with the methodology of this work it is proceeded as follows:

- In this first stage is calculated the AR part of the model. For this, from $K = 2$ are built the models AR with K terms and is tested the performance on the test set. As soon as the first K value is obtained where the RSS of the model is less than the values obtained for the $K - 1$ and $K + 1$ is considered that the AR part of the model has the already found K terms and passes to the second stage.
- It is calculated the error series obtained from the original series and the ones calculated by the model obtained in the previous stage. On this new series it is applied the same procedure above mention and it is obtained the part corresponding to the component of the MA moving average of the ARMA model. It may be the case that by including the MA components of the model it will be had the worst approximations in the test set than those obtained with the AR part. In this case the model would only have the AR component.
- It is checked if the model AR obtained in the stage 1 presents the FD phenomenon occurs, and if so to realize the displacement of the graph one unit to the left according to (9) as long as with this procedure the result is improved.

To test the performance of our models of (8) we used the series A, B, C, D, E and F appearing in (Box & Jenkins, 1976), used and presented in chapter 3.

In (Hansen at all, 1999) are shown the results of building several linear models for these series. The first is the classic BJ, and others apply when BJ model do not satisfy the postulate that the error is a white noise. In (McDonald & Yexiao, 1994) it is indicated that the use of these latest models improved from 8% to 13% the capability of prediction of the model when the error is not white noise. Immediately it is presented the relationship of these models for the linear models.

- Standard ARIMA model. Here applies the traditional methodology of BJ where the main components are the autoregressive models with moving averages that are linear in the time series $\{Z_t\}$ and white noise $\{a_t\}$ (Box & Jenkins, 1976).
- Ordinary least squares (OLS). These are used when the distribution of the error presents the leptokurtosis problem and allows diminishing the error in the forecasting (Huber, 2004).
- Least Absolute Deviation (LAD). It is used to minimize the sum of the absolute values rather the sum of squares. This is done to reduce the influence of the extreme errors (Huber, 2004).
- Generalized t-distribution (GT). Here is minimized the objective function in relation to the parameters but assuming that the error has a t-distribution (McDonald & Newey, 1988).
- Exponential Generalized beta distribution of the second kind (EGB2). Here it is supposed that the errors have a distribution of this kind (McDonald & Newey, 1988).

Additionally in (Hansen at all, 1999) are presented the results of two models of neural networks, one heuristic (Heuristic NN), and another based on genetic algorithms (GANN), which are included in the commercial software BioComp System's NeuroGenetic Optimizer[®].

To make comparisons with the models described above, it will be used the same size of training set and test sets shown in (Hansen at all, 1999), where if the number of elements of the series is greater than 100 the sizes of test sets are set to 10. In the event that they are less than or equal to 100 the test sets will have size five. The size of the training sets is the original size of the series minus the number of elements of the test set.

With the methodology of this work were obtained the models of the Table 3, where for each example is presented the component AR and if necessary the MA. Note that when it is shown "AF" in the last column of the table it was applied the displacement presented in (9).

Series	AR	MA	AF
A	$1.1035 + 0.5648_1 + 0.1919_6 + 0.1245_9 + 0.0544_{13}$	$-0.2271_1 + 0.1046_2 + 0.051_{43}$	No
B	$0.8302 + 1.1274_1 - 0.1685_2 + 0.0644_4 - 0.0258_6$	$0.0460_2 - 0.0576_5 + 0.1381_6 + 0.0587_7$	Yes
C	$0.8425_1 - 0.8488_2$		No
D	$0.7609 + 0.8997_1 + 0.0511_{12} - 0.0335_{16}$		Yes
E	$1.9993 + 1.0051_1 - 0.2590_3 + 0.1538_{10}$		Yes
F	$1.9996 + 0.6555_2 + 0.2938_3$		No

Table 3. Solution to the Box Jenkins problems.

In Table 4 are shown the results of the different methodologies presented in (Hansen at all, 1999) and those obtained with the algorithm proposed in this work. Table 4 is used as a criterion of comparison of the sum of absolute values of errors. The results of our model are presented in the line called "Linear AF" and the place obtained when confronted with other models is in the line called "Place." It should be noted that each group of comparisons, except in one instance", the results obtained with our methodology are better than those obtained with the confronted statistical methods and also have good results when compared with those obtained by neural networks.

Table 5 presents the results of comparing the method proposed in this work with those reported in (Cortez at all, 2004). In this paper are confronted the methodologies:

- Holt-Winters Methodology. This methodology is widely used due to its simplicity and accuracy of its forecasting's especially with periodic time series. It is based on four basic equations that represent the regularity, trend, periodicity and forecasting of the series (Chatfield, 2000).
- Box-Jenkins Methodology that already was widely commented in previous sections (Box & Jenkins, 1976).
- Evolutionary forecasting method. It is a methodology based on evolutionary programming (Cortez at all, 2004).
- Evolutionary meta algorithms. It is a metaheuristic that uses two architecture levels, in the first is chosen the ARMA model in question, and in the second the corresponding parameters are estimated (Cortez at all, 2004).

To test the performance of the models, were used some of the series in (Hyndman, 2003). which are known as: Passengers, which is a series (144 data) that represents the number of monthly passengers on an airline; Paper, this series (120 data) represents the paper monthly sales in France; Deaths, which is a series (169 data) that represents the death and injury on roads of Germany; Maxtemp represents the maximum temperatures (240 data) in Melbourne, Australia; and Chemical, which is a series (198 data) of readings of the concentrations of a chemical reactor. The training sets of these series contain 90% of the data and remaining 10% are in the test set.

	Series A	Series B	Series C	Series D	Series E	Series F	Series G
Linear AF	3.9	75	5.9	2.87	87	46	173
Heuristic NN	4.519	88.312	9.138	2.942	98.873	43.966	
GA NN	3.705	72.398	6.684	2.952	69.536	36.4	
ARIMA ML	4.005	78.855	11.247	3.114	3.114	49.161	
OLS	3.937	83.17	10.74	3.08	114.8	45.5	
LAD	3.96	79.47	10.3	3.066	117.6	44.46	
GT	3.937	80.68	10.25	3.064	106.5	44.59	
EBG2	4.017	81.01	10.3	3.066	111.8	44.5	
PLACE	1	2	1	1	2	6	

Table 4. Comparison of the models with regard to a sum of values of absolute errors.

Using the method proposed in this work it were obtained the models that are shown in table 5. Note that form this examples none presents DF.

In Table 6 were confronted the results for these TS. The results of our models are shown in the column called "Linear AF" and the place gotten when comparing with the other models is shown in the column "Place".

From the results presented in the tables of this section it can be concluded that the model built with our methodology outperform all the models obtained with statistical methods and are competitive with non-linear methods presented here. In addition, it must be added that this methodology is fully automated and allows modelling TS than other traditional methodologies can not.

Series	AR	MA
Passengers	$1.3400 + 0.9087_1 + 1.0612_{12} - 0.9633_{13}$	$4.3636 + 0.405_2$ $18.662 + 0.1045_5 - 0.1857_{11}$ -0.1930_{12}
Paper	$6.2323 + 0.9583_{12}$	
Deaths	$1.9941 + 0.9053_{12} + 0.0832_{14}$	
Maxtemp	$0.7046 + 0.3362_1 - 0.0668_7 + 0.4060_{11}$ $+0.2910_{12}$	
Chemical	$0.5419 + 0.6081_1 + 0.3753_7 - 0.0144_{15}$	

Table 5. Solutions with the methodology proposed in this work.

Series	Holt Winter	Box Jenkins	Heuristic Evolutionay	Meta Evolutionay	Linear AF	Place
Passengers	16.5	17.8	21.9+ -1.2	17.2+ - 0.2	16.3	1
Paper	49.2	61	60.2+ - 2.2	52.5+ - 0.1	5.59	1
Deaths	135	144	135.9+ - 1.7	137+ - 2	140	3
Maxtemp	0.72	1.07	0.95+ - 0.02	0.93+ - 0.4	0.94	2
Chemical	0.35	0.36	0.36+ - 0.0	0.34+ -0.0	0.34	1

Table 6. Comparison with other methodologies.

10. Conclusions

From the above it can be obtained several conclusions. The first is that the methodology developed here based on setting out the building of linear models as an optimization problem, where the construction of the problem is guided by the classical TS theory, is correct because allows to build better models than those obtained by the traditional methods.

Another conclusion is that the fact of choosing the SAGA as an alternative to solve the problems set out here is very important since allows exploring the solution space of our problem and finding the most significant variables to solve it. In addition, the SAGA version developed has proved to be very robust in solving many different problems with out adjustment of parameters.

As a result not contemplated it was found that the phenomenon of FD, which allowed us to construct new linear models for TS, which in some cases are better alternatives compared to other linear and nonlinear models. In addition, these new models have great potential for application in areas such as industrial control, economics, finance, etc. In particular, we think that the FD is a characteristic of the phenomenon in question, but that is only detected if the model is built with an appropriate methodology, particularly in the selection and setting limits of variables.

Finally, it should be noted that having a fully automated methodology with the ability to model phenomena that other methodologies can not open a whole world of possibilities in the development of computer systems for modelling and process control.

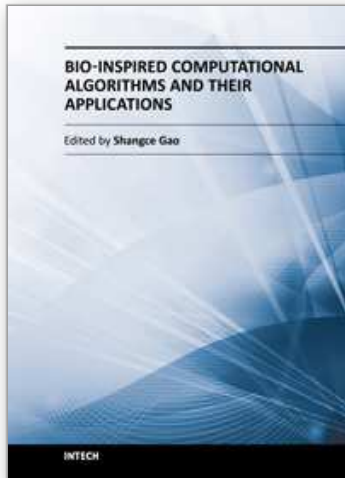
11. References

- Alberto I. & Beamonte A. & Gargallo P. & Mateo P. & Salvador M.(2010). Variable selection in STAR models with neighbourhood effects using genetic algorithms. *Journal of Forecasting*, Vol 29, Issue 8, page numbers (728-750), ISSN 0277-6693.

- Bäck T. (1992). The interaction of mutation rate, selection, and self-adaptation within genetic algorithm. *Proc. 2nd Conf. on Parallel Problem Solving from Nature*, (Brussels,1992), ISBN 0444897305, Elsevier Amsterdam.
- Bäck T. (1992). Self-adaptation in genetic algorithms. *Proc. 1st Eur. Conf. on Artificial Life*. MIT Press. Cambridge, MA.
- Battaglia F. & Protopapas M.(2011). Time-varying multi-regime models fitting by genetic algorithms. *Journal of Time Series Analysis*, Vol 32, Issue 3. page numbers (237-252), ISSN 1467-9892.
- Box G. & Jenkins G. (1976). *Time Series Analysis: Forecasting and Control*.Holden-Day, INC. ISBN 0-13-060774-6, Oakland, California USA 1976.
- Chatfield C. (2000). *The Series Forecasting*. CRC Press. ISBN 1584880635, USA.
- Chiogna M. & Gaetan C. & Masarotto G.(2008). Automatic identification of seasonal transfer function models by means of iterative stepwise and genetic algorithms. *Journal of Time Series Analysis*, Vol 29, Issue 1, page numbers (37-50), ISSN 1467-9892.
- Cortez P. & Rocha M. & Neves J. (1996). Evolving Time Series Forecasting ARMA Models. *Journal of Heuristics*, Vol 10., No 4., page numbers (415-429), ISSN 1381-1231.
- Eiben Á. E. & Hinterding R. & Michalewicz Z. (1999). Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, Vol 3., No 2., page numbers (124-141), ISSN 1089-778X.
- Flores P. & Garduño R. & Morales L. & Valdez M. (1999). Prediction of Met-enkephalin Conformation using a Sieve Self Adaptive Genetic Algorithm. *Proceedings of the Second International Symposium on Artificial Intelligence: Adaptive Systems ISSAS'99.*, page numbers (186-190)
- Garduño R. & Morales L. & Flores P. (2000). Dinámica de Procesos Biológicos no Covalentes a Nivel Molecular. *Revista Mexicana de Física*, Vol 46., Suplemento 2., page numbers (135-141), ISSN 0035-001X.
- Garduño R. & Morales L. B. & Flores P. (2001). About Singularities at the Global Minimum of Empiric Force Fields for Peptides. *Journal at Molecular Structure (Theochem)*, page numbers (277-284), ISSN 0166-1280.
- Guerrero V. (2003). *Análisis Estadístico de Series de Tiempo Económicas*,Thomson Editores, ISBN 9706863265, México DF.
- Hansen J. & McDonald J. & Nelson R. (1999). Time Series Prediction with Genetic Algorithms designed neural networks, an empirical comparison with modern statistical models. *Computational Intelligence*, Vol 15., No 3., page numbers (171-184), ISSN 0824-7935.
- Huber P. (2004). *Robust Statistics*. Wiley-IEEE. ISBN 978-0-521-88068-8, USA.
- Hyndman R.& Koehler A. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, Vol 22, Issue 4, page numbers (679-688), ISSN 0169-2070.
- Hyndman R. (2003). Time Series Data Library. Available in <http://www.robjhyndman.com/TSDL>
- Mateo F. & Sovilj D. & Gadea R.(2010). Approximate k -NN delta test minimization method using genetic algorithms; Application to time series. *Neurocomputing*, Vol 73, Issue 10-12. page numbers (2017-2029), ISSN 0925-2312.
- McDonald J. & Newey W. (1988). Partially adaptive estimation of regression models via the generalized t distribution. *Econometric Theory*, Vol 4., page numbers (428-457), ISSN 0266-4666.
- McDonald J. & Yexiao X. (1994). Some forecasting applications of partially adaptive estimators of ARIMA models. *Economics Letters*, Vol 45., Issue 4., page numbers (155-160), ISSN 0165-1765.

- Medeiros C. & Veiga Á. (2000). A Hybrid Linear-Model for Time Series Forecasting. *IEEE Transactions on Neural Networks*, Vol 11., No 6., page numbers (1402-1412), ISSN 1045-9227.
- Medeiros C. & Veiga Á. (2005). A Flexible Coefficient Smooth Transition Time Series Model. *IEEE Transactions on Neural Networks*, Vol 16., No 1., page numbers (97-113), ISSN 1045-9227.
- Miller B. & Goldberg D. (1995). Genetic Algorithms, Tournament Selection, and the Effects of Noise. *Complex Systems*, Vol 9., page numbers (193-212), ISSN 0891-2513.
- Journal of Time Series Analysis. available in <http://www.blackwellpublishing.com/journal.asp?ref=0143-9782>
- Palit A. & Popovic D. (2005). *Computational Intelligence in Time Series Forecasting, theory and engineering applications*. Springer-Verlag. ISBN 1852339489, London.
- Sánchez V. & Flores P. & Valera J. & Pérez M. (2004). Mass Balance Calculations in Copper Flash Smelting by Means of Genetic Algorithms. *Journal of Metals*, Vol 56., No 12., page numbers (29-32), ISSN 1047-4838.
- Szpiro G. (1997). Forecasting chaotic time series with genetic algorithms. *Physical Review E*, Vol 5., No 3., page numbers (2557-2568), ISSN 1539-3755.
- Weigend A. & Gershenfeld N. (1994). *Time Series Prediction, Forecasting the future and Understanding the Past*. Addison-Wesley Publishing Company. ISBN 9780201626025, USA.
- Yadavalli V. & Dahule R. & Tambee S. & Kulkarni B. (1999). Obtaining functional form for chaotic time series using genetic algorithm. *CHAOS*, Vol 9., No 3., page numbers (789-794), ISSN 1054-1500.

IntechOpen



Bio-Inspired Computational Algorithms and Their Applications

Edited by Dr. Shangce Gao

ISBN 978-953-51-0214-4

Hard cover, 420 pages

Publisher InTech

Published online 07, March, 2012

Published in print edition March, 2012

Bio-inspired computational algorithms are always hot research topics in artificial intelligence communities. Biology is a bewildering source of inspiration for the design of intelligent artifacts that are capable of efficient and autonomous operation in unknown and changing environments. It is difficult to resist the fascination of creating artifacts that display elements of lifelike intelligence, thus needing techniques for control, optimization, prediction, security, design, and so on. Bio-Inspired Computational Algorithms and Their Applications is a compendium that addresses this need. It integrates contrasting techniques of genetic algorithms, artificial immune systems, particle swarm optimization, and hybrid models to solve many real-world problems. The works presented in this book give insights into the creation of innovative improvements over algorithm performance, potential applications on various practical tasks, and combination of different techniques. The book provides a reference to researchers, practitioners, and students in both artificial intelligence and engineering communities, forming a foundation for the development of the field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Pedro Flores, Larysa Burtseva and Luis B. Morales (2012). Self Adaptive Genetic Algorithms for Automated Linear Modelling of Time Series, Bio-Inspired Computational Algorithms and Their Applications, Dr. Shangce Gao (Ed.), ISBN: 978-953-51-0214-4, InTech, Available from: <http://www.intechopen.com/books/bio-inspired-computational-algorithms-and-their-applications/self-adaptive-genetic-algorithms-for-automated-linear-modelling-of-time-series>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen