

# Automated Linear Modeling of Time Series with Self Adaptive Genetic Algorithms

Pedro Flores, Carlos Anaya, Héctor M. Ramírez, and Luis B. Morales.

**Abstract**—In this paper we present two algorithms that automatically calculate linear expressions for Time Series. To estimate the maximum number of terms of the linear expression and the intervals in which the series coefficients vary, the algorithms are based in the Box-Jenkins methodology. With this information and establishing beforehand the number of terms that are required, the Self Adaptive Genetic Algorithms are applied in several stages to obtain the series model. The proposed algorithms were tested in the Box-Jenkins classical examples, obtaining satisfactory results. It is worth it to mention that these algorithms allow treating series with time-dependent trends and variances. The methodology based on Self Adaptive Genetic Algorithms is used to estimate linear models for every example of NN3 2007, although in this paper we are presenting only the results of NN3-REDUCED.

## I. INTRODUCTION

TIME SERIES (TS) are models that allow the forecasting of the behavior of real data. In the specific case of linear TS, the value in a moment shall be estimated as a linear expression of the values of a certain number of former measurements such as in (1). Linear models of TS are important because there are numerous applications where linear estimations are enough [13], [14], in addition that they are widely used in industrial applications. The most classical reference for the treatment of linear models is [3]. Additionally, other methodologies use them to make forecasts [4], [15], [16]

First of all, the problem in finding a good linear model for a series of data is that it requires the determination of how many and which terms are the most appropriate to solve this problem. Secondly, it is necessary to know in which intervals the coefficients of the linear expression are, and finally, to find the values for these coefficients that minimize the quadratic error for the whole history. In this way we have to consider a problem of non-linear optimization with

P. Flores is working at the Department of Mathematics, Universidad de Sonora, Blvd. Luis Encinas y Av. Rosales S/N Col. Centro, Hermosillo, Sonora. México. 83000. (corresponding author to provide phone: 526622592155, fax: 526622592219, e-mail: pflores@gauss.mat.uson.mx)

C. Anaya and H. M. Ramirez are working at the Department of Industrial Engineering, Universidad de Sonora, Blvd. Luis Encinas y Av. Rosales S/N Col. Centro, Hermosillo, Sonora. México. 83000. (e-mail: canaya@industrial.uson.mx, hramirez@industrial.uson.mx).

L. B. Morales is working at the Instituto de Investigación en Matemáticas Aplicadas y Sistemas, Universidad Nacional Autónoma de México. Ap. Postal 70-221, México. DF. 04310 (e-mail: lbm@servidor.unam.mx).

variables in real intervals, whose limits shall be specified. This problem has multiple local minimums; therefore, proper optimization techniques are required.

To address this problem in this work, we first applied the main results of the TS statistics analysis [3], [10] in order to estimate which is the greatest number of terms that will appear in the series, and the intervals in which they vary.

Next, in order to estimate the best coefficients of the linear expression, we used, in several stages, Genetic Algorithms (GA), which are techniques of optimization that implement global multi-point search, quickly locating areas of high quality. To have a GA successful implementation, we require estimating the value of certain parameters, which is done through several tests. In this work, we implement versions of Self Adaptive Genetic Algorithms (SAGA) because with these versions we eliminate the stage of repetitive tests that are required to adjust the parameters, thus achieving that the same version of the code solves any problem of TS linear modeling without the participation of the user. Regardless of the existence of papers where GA are used in TS like [5],[6],[11],[21],[22], it is important to mention that any reference of the use of SAGA in the area was found.

The plan in this paper is the following. In the second section, the results of the Box-Jenkins (BJ) methodology that we use in our proposal are summarized. In the third section, we present the characteristics of SAGA that we are going to use. In the fourth section, we present the characteristics of the algorithms that we developed. Following, in the fifth section, we summarize the results obtained when we solved the classical examples that appear in [3]. In the sixth section we present the results of modeling the examples of the NN3 2007 reduced version; and finally, in the seventh section, we present the conclusions of this work.

## II. MAIN RESULTS OF THE BOX-JENKINS METHODOLOGY

Univariate time series were analyzed using BJ methodology from the construction of equations in differences with an adding randomized component called white noise. From these models, the conditions under which the series have the stationarity property and the scheme that shall be followed to estimate the parameters of the model in particular are determined. The most general model is called ARMA (p,q), which indicates the presence of autoregressive components in the observable variable,  $z_t$ , and in the white

noise variable  $a_t$ . A particular kind of model for stationary series corresponds to the autoregressive models (of  $p$  order) whose representation is given by the expression.

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

where  $\delta$  y  $\phi_j$  are constants that meet the following equations:

$$|\phi_j| < 1, \quad \mu = \frac{\delta}{1 - \sum \phi_j}. \quad (2)$$

and

$$\sum \phi_j < 1$$

$\mu$  represents the series average. Equations (1) and (2) are the consequence of the stationary property.

The correlation structure presented by a time series, associated to an autoregressive model for observations separated  $k$  units of time is given by the following autocorrelation function:

$$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2} + \dots + \phi_p \rho_{k-p}$$

Where  $\rho_k$  represents the autocorrelation for data of the series separated  $k$  units of time. From initial conditions that meet this equation in differences, the following possible behaviors are presented: exponential decay or sinusoidal decay. The former allows determining whether a series is stationary or not.

Methodology of BJ meets the following stages:

a) Identification of a possible model within the kind of ARIMA models. Thus, it is determined whether the series is stationary or not. In the case of an observed series is not stationary, the Operator Difference  $\nabla z_t = z_t - z_{t-1}$  is applied the times that are necessary to reach stationarity. To avoid over differentiation, the variance of the new series is obtained selecting the lowest variance. In case a series is stationary in the middle level, but its dispersion increases or decreases, then, a transformation, generally logarithmic, shall be applied in order to stabilize the variance.

Given a stationary series, the behavior pattern of the autocorrelation function and the partial autocorrelation function indicates the possible number of  $\phi$  parameters that the model shall have.

In addition to the presence of stationarity in a temporary series, another property is required in the ARIMA models, which is called invertibility; this property allows the representation of the series as an autoregressive model of infinite extension that meets the following condition:

$$\lim_{j \rightarrow \infty} \phi_j = 0$$

The former allows that taking a finite number of terms, a satisfactory expression of the form (1) for the series is obtained.

b) Estimation of the parameters involved in the model, through non-linear estimation techniques.

c) Verification that the model provides an adequate adjustment and that the basic suppositions implicit in the model are met.

d) Use of the model.

In this paper, the TS of data that appear in [3] with the name of Series A, B, C, D, E, F, and G were used. From these models, when treating the G series with the BJ methodology, a linear expression of the (1) form is not found, and, in addition, it presents growing variance; therefore, it is necessary to apply a logarithmic transformation to stabilize its variance. It is important to mention that with our proposal all the former problems can be satisfactorily solved.

### III. SELF ADAPTIVE GENETIC ALGORITHMS.

John Holland developed GA [12], which was inspired by the basic principles that rule the evolution of the species. They have been successfully used in the resolution of many problems [9],[17], and in order to apply them we require: a genetic representation of the individual, a way to create an initial population, a function of performance that revises how adapted to the environment the individual is, and he is separated by a procedure called selection according to its performance, the genetic operators that alter the structure of population and the value of the various parameters that are required by the algorithm.

In our proposal the individuals will be vectors of real components. The initial population will be created at random. The performance functions in every stage of the aforementioned will be the functions to optimize according to the problem in question. In this work, the function Root of the Sum of Square Errors (RSSE) is used, but this can be changed by other function. The tournament selection [18] is the procedure of selection that will be used; additionally, in order to avoid the problem of premature convergence, it is established that in every population only ten copies of the best individual can be. Multiple crossing and mutation will be the genetic operators. The parameters that we are going to use will be the size of the population, probability of individual crossing, repetition of crossing, probability of individual mutation, and repetition of mutation. The aforementioned is necessary because in our model the probability of crossing and mutation will be characteristics of every individual (not of the population as in the GA), and moreover, it is considered that crossing and mutation can be multiple, that is to say, they can act several times in the same individual. Individual crossing will be executed interchanging the components and the mutation of a component of an individual that will be carried out changing

the value of the individual crossing between set limits at random.

SAGA algorithms were developed by Thomas Back [1],[2] and have the characteristics that they automatically look for the best parameters for their operation. In our case, the population size will be fixed and equal to one hundred, and we will use the following four auto adaptable parameters: the probability of individual crossing that varies in the interval [0.5,0.95], repetition of crossing in [1.0, 4.0], probability of individual mutation that varies in [0.5,0.85], and repetition of mutation in [1.0,5.0]. The individuals in our problems will be proposals of solution for our problems and, additionally, they will have four more components where the values of the auto adaptable parameters are represented. The way to operate with these parameters is similar to the way presented in [1],[2], taking the respective averages as the case may be. A binary version of this algorithm was successfully used by one of the authors of this paper in other problems [7],[8],[16],[19], and, according to the bibliographic review, this version is original.

Main disadvantage with the utilization of SAGA is the higher computer cost as compared to traditional versions, but the gain that we have is that with the same code we can automatically solve every problem of the linear modeling of the NN3 series, without the need to adjust the parameters for every one.

Regarding the way in which SAGA algorithms are going to be used, we can group them in three stages. In the first stage, they will be used to explore the space of solutions and to determine which variables are the most important for the problem in question. In the second stage, they will be applied to find solutions taking into account only the variables that are important for the problem, and finally, in the third stage, the best solutions will be tuned up. It may be observed that in every one of the stages, the target function to minimize is RSSE and what it changes in every stage is the variable to consider and the limits of those variables.

#### IV. PROPOSED ALGORITHMS.

The proposed algorithms are developed taking into account the RSSE as the function to minimize, but they can be extended to use another measure of approximation among the series.

The first algorithm that we built was based on the BJ methodology, and for the first stage, what we first did was to decide with which series we were going to work, among the original series, the first differences, the second differences, or, in our case, we included the possibility to work with third differences. To decide this, we chose the one with the minimum variance to be sure that it will be a stationary series. Once we have decided with which series we were going to work, we estimated how many terms are necessary for the linear approximation of the chosen series considering the autocorrelation function. Thus, we consider every term between two consecutive maximums of this function. With

the former information, we built the limits for the intervals of the coefficients of the chosen series for which we took all of them in [-1,1], except the independent term whose limits were estimated between zero and the average value of the series. The motivation to establish these limits is obtained from the equations (2). Later on, we made ten repetitions of five hundred iterations each one of SAGA, and with every result we built a solution averaging the sum of the absolute values of the components of the ten results. This gave us information about the most important components because these ones will be the ones that have the greatest values in the solution.

For the second stage, the components of the former solution are ordered in descending order, and a certain numbers established beforehand are chosen (generally, for our case is four). With the chosen components and the same limits established beforehand, five executions of SAGA are carried out, taking as stop criterion that in the former two hundred and fifty iterations the optimum did not change, and from this five executions, the best result is taken.

For the last stage, a solution for the original series is built with the former result, and with this result new limits are built, taking every variable of the solution in a vicinity of radius one. Here we performed another five SAGA executions similar to the ones of the former paragraph, and finally, the best result of these five executions is taken, and the very small terms are discarded. It should be noted that in this case, it is possible that the solutions we get have more than four terms.

The second algorithm uses the BJ methodology only to estimate how many terms are necessary for the linear approximation of the stationary series, and with this, the number of terms that are used for the approximation of the original series are estimated. From now on, the stages of the former algorithm are repeated taking the limits of all the coefficients in [-1,1], and always working with the original series. There is not a result that justifies the use of these limits, except the fact that in some cases the results obtained are better than the ones obtained with the initial algorithm. As an example of this we have that in five of the eleven examples of NN3-REDUCED this algorithm had better performance than the one formerly presented.

#### V. COMPARITION WITH BOX-JENKINS.

To be able to properly compare our work with the examples treated by BJ, we took the results of the models of Table 7.13 on page 239 from [3], and applying the property of invertibility we proceeded to obtain expressions of the form (1) that appear in TABLE I.

In that table, are the expressions for the model estimated by our algorithms (OA) when we ask to find a linear model with the same number of terms and always choosing the best result. Additionally, we present the values of RSSE for both models. From these, we conclude that the results are similar, but with the advantage that our algorithms are totally

automated and they do not require the application of a previous stage of data pre-processing. Next, we present the results of NN3-REDUCED.

TABLE I  
COMPARATION WITH BOX-JENKINS

Serie	Linear Models	RSSE
A-BJ	$z_t = 0.3z_{t-1} + 0.21z_{t-2} + 0.147z_{t-3} + 0.1029z_{t-4} + 0.072z_{t-5} + 0.0504z_{t-6} + 0.0348z_{t-7}$	20.003
A-OA	$z_t = 0.9264 + 0.3705z_{t-1} + 0.2850z_{t-2} + 0.0783z_{t-4} + 0.2126z_{t-4}$	4.278
B-BJ	$z_t = 1.09z_{t-1} - 0.0981z_{t-1}$	155.365
B-OA	$z_t = -0.2799 + 1.000z_{t-1} + 0.1242z_{t-1} - 0.1242z_{t-17}$	135.128
C-BJ	$z_t = 1.82z_{t-1} - 0.820z_{t-1}$	1.989
C-OA	$z_t = 1.8136z_{t-1} - 0.8140z_{t-2}$	1.984
D-BJ	$z_t = 1.17 + 0.87z_{t-1}$	5.220
D-OM	$z_t = 1.3343 + 0.8544z_{t-1}$	5.256
E-BJ	$z_t = 14.35 + 1.42z_{t-1} - 0.73z_{t-2}$	149.137
E-OM	$z_t = 1.7618z_{t-1} - 1.2471z_{t-2} + 0.4200z_{t-3}$	155.292
F-BJ	$z_t = 58.87 - 0.340z_{t-1} + 0.190z_{t-2}$	85.015
F-OM	$z_t = 52.1273 - 0.2577z_{t-1} + 0.2416z_{t-2}$	85.085
G-OM	$z_t = 1.9991 + 0.2199z_{t-1} + 0.1472z_{t-11} + 0.7183z_{t-12}$	215.354

In Fig. 1. we show the graphic of Serie G.

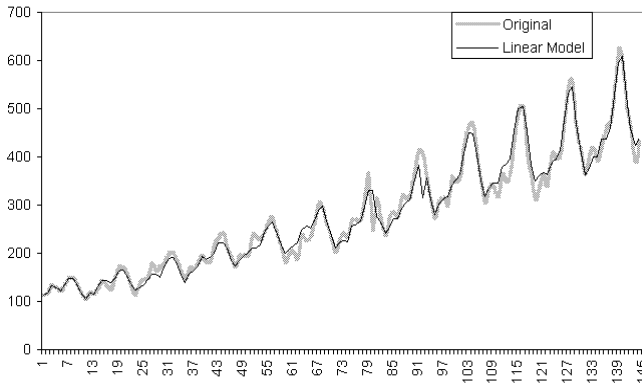


Fig. 1. Serie G

## VI. RESULTS OF NN3-REDUCED.

With the two algorithms that we developed, we perform fifty executions of every one in every example of NN3-REDUCED, looking for linear models with four terms. In TABLE II we show the results of the linear expressions as well as the estimation of RSSE.

After revising the behavior of the fifty solutions of these examples, we concluded that we only need five executions to obtain satisfactory results. Therefore, for the examples of the NN3 complete series, we only carried out five executions with every algorithm and chose the best of them. The results of all NN3 examples are not presented in this work.

TABLE II  
LINEAR MODELS FOR NN3-REDUCED

Problem	Lineal Model	RSSE
101	$z_t = 1269.3358 + 0.3467z_{t-1} + 0.6978z_{t-12} - 0.2921z_{t-13}$	1713.550
102	$z_t = 1.9987 + 0.9218z_{t-1} + 0.9574z_{t-12} - 0.8792z_{t-13}$	5440.262
103	$z_t = 1.9989 + 0.5152z_{t-1} + 0.8124z_{t-12} - 0.3734z_{t-13}$	80019.738
104	$z_t = 9.113 + .7252z_{t-1} + 0.8316z_{t-12} - 0.5592z_{t-13}$	7321.538
105	$z_t = 1.998 + 0.9099z_{t-1} + 0.3104z_{t-11} - 0.2225z_{t-13}$	1513.984
106	$z_t = 2821.9541 + 0.2673z_{t-2} - 0.1699z_{t-7} + 0.3422z_{t-12}$	4464.870
107	$z_t = 0.9978 + 0.7937z_{t-1} + 0.3152z_{t-12} - 0.1125z_{t-13}$	1387.011
108	$z_t = 2000.5819 + 0.2885z_{t-2} - 0.1456z_{t-4} + 0.2379z_{t-5}$	10417.433
109	$z_t = 1.9988 + 0.9951z_{t-1}$	2297.306
110	$z_t = 1863.0699 + 0.252z_{t-1} - 0.10582z_{t-5} + 0.2379z_{t-11}$	18593.279
111	$z_t = 474.1106 + 0.242z_{t-11} - 0.3319z_{t-12} + 0.2688z_{t-13}$	7248.281

On the other hand is important to state that in some cases the solution of the other algorithm has better visual behavior than the optimum (see Fig. 2. and 3.), as in the example 110 of NN3.

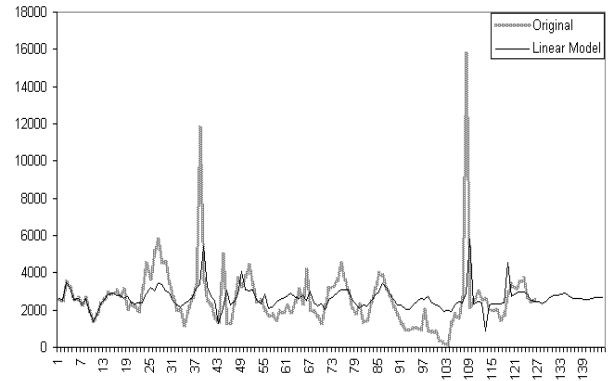


Fig. 2. Example 110 with the first Algorithm (RSSE=18593.279)

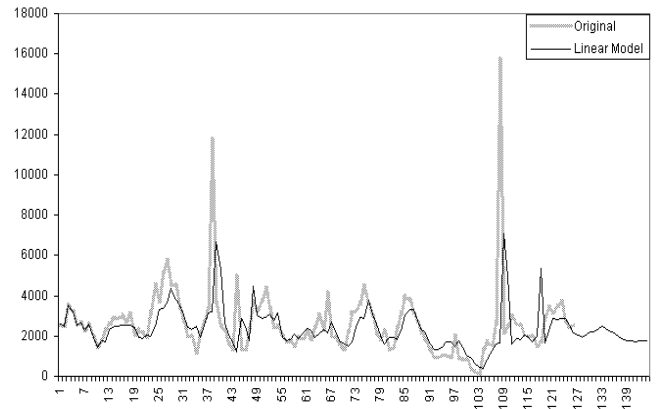


Fig. 3. Example 110 with the second Algorithm (RSSE=20055.177) and Linear Model  $z_t = 1.9999 + 0.3920z_{t-1} + 0.2498z_{t-2} + 0.2706z_{t-3}$

In Fig. 4. to Fig. 13. we show the graphics of results.

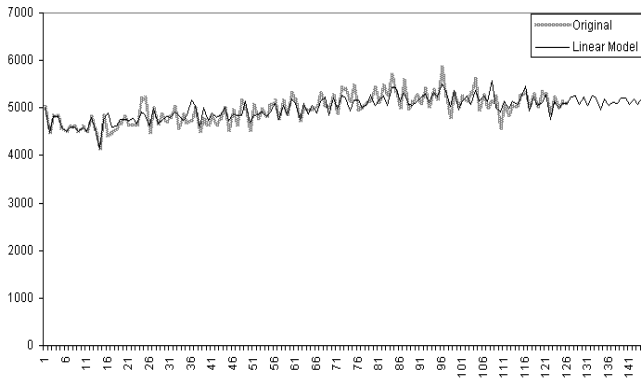


Fig. 4. Example 101

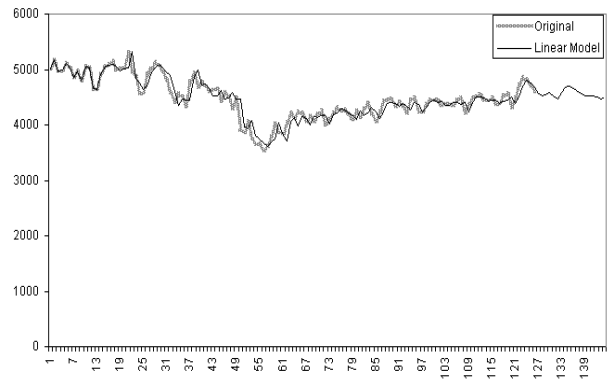


Fig. 8. Example 105

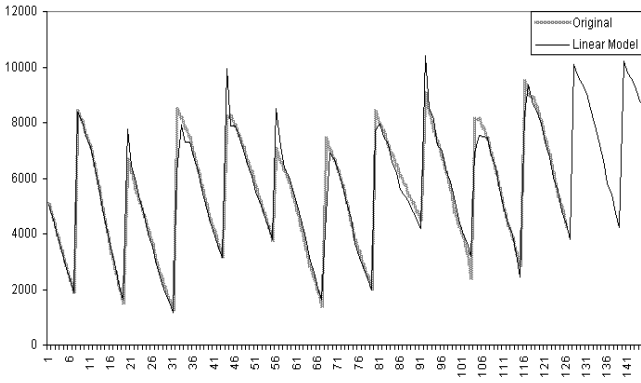


Fig. 5. Example 102

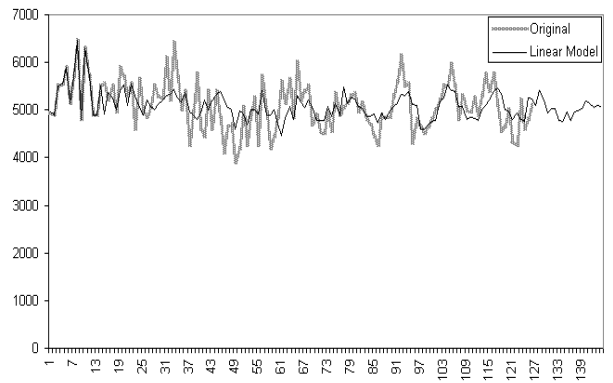


Fig.9. Example 106

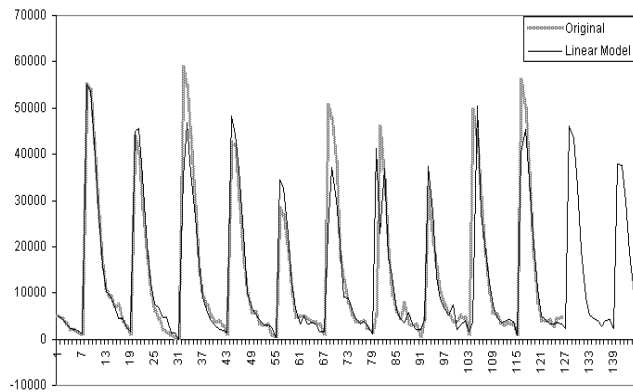


Fig. 6. Example 103

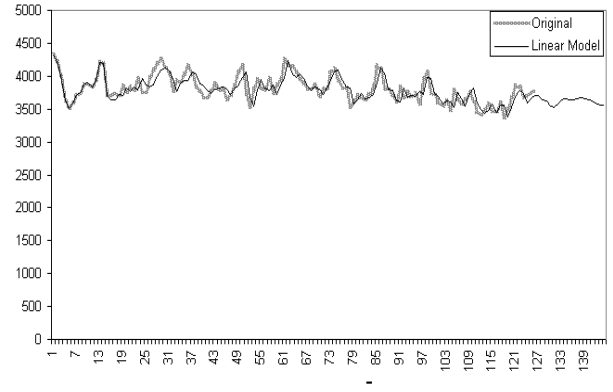


Fig. 10. Example 107

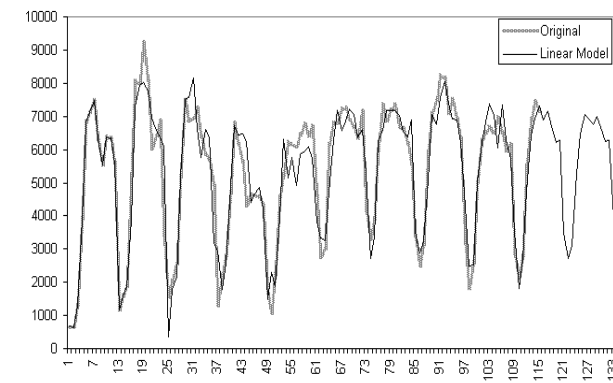


Fig. 7. Example 104

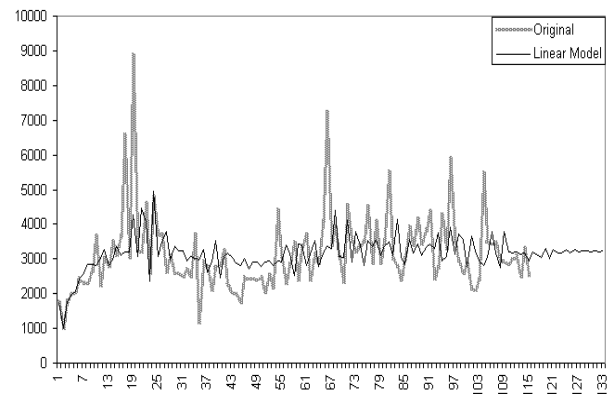


Fig. 11. Example 108

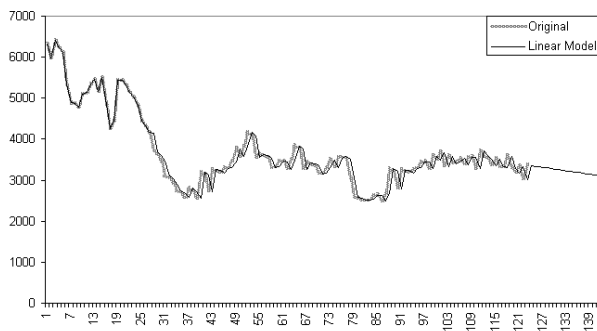


Fig. 12. Example 109

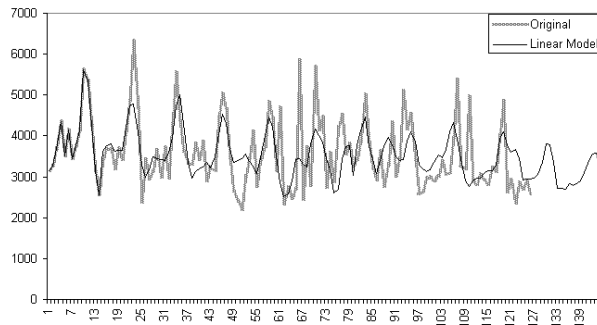


Fig. 13. Example 111

## VII. CONCLUSIONS.

In this work we built two algorithms that allow the fully automated construction of TS linear models, with a certain number of terms set beforehand with satisfactory results. Moreover, our proposal allows treating series that show trends and variances that change with time, which in general the BJ methodology cannot treat.

The fact that we can establish beforehand the number of terms that are required in the linear approximation allows better versatility in the accuracy of the approximation, because when the number of terms is greater, the accuracy obtained is greater too. On the other hand, the fact that we have solutions from two algorithms allows us to choose from the two approximations to the series the one that is better for our interests.

The efficiency and accuracy of these algorithms can be improved in the future, carrying out tests to establish smaller intervals of auto adaptable parameters and modifying the final stage applying some local optimization algorithm based in the trajectories.

We are planning to investigate in the future under which theoretical conditions we can conclude that the second algorithm has better behavior than the first one.

## REFERENCES

[1] Bäck Thomas. "The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm". *Proc. 2<sup>nd</sup> Conf. on Parallel Problem Solving from Nature* (Brussels, 1992) Elsevier Amsterdam. 1992.

[2] Bäck Thomas. "Self-adaptation in genetic algorithms". *Proc. 1st Eur. Conf. on Artificial Life*. MIT Press. Cambridge, MA. 1992.

[3] Box George E., Jenkins Gwilym M. *Time Series Analysis: Forecasting and Control*. Holden-Day, INC. Oakland, Calif. USA 1976.

[4] Carvalho Alexandre X., Tanner Martin A. "Mixtures-of-Experts of Autoregressive Time Series: Asymptotic Normality and Model Specification". *IEEE Transactions on Neural Networks*. Vol. 16, No. 1. January 2005. Pp. 39-56.

[5] Cortez Paulo, Rocha Miguel, Neves Jose. "Evolving Time Series Forecasting ARMA Models". *Journal of Heuristics*. Vol. 10. No. 4. July 2004. Pp.415-429.

[6] Gaetan Carlo. "Subset ARMA Model Identification Using Genetic Algorithm". *Journal of Time Series*. Vol. 21. Issue 5. Pp.559-570. September 2000.

[7] Garduño Ramon, Morales Luis B., Flores P. Pedro. "Dinámica de Procesos Biológicos no Covalentes a Nivel Molecular". *Revista Mexicana de Física*. Vol 46. Suplemento 2. Pp. 135-141. 2000.

[8] Garduño Ramon, Morales Luis B., Flores P. Pedro. "About Singularities at the Global Minimum of Empiric Force Fields for Peptides". *Journal at Molecular Structure (Theochem)*. 543 . 277-284. 2001.

[9] Goldberg D. E. *Genetic Algorithms in Search, optimization and Machine Learning*. Addison Wesley, Reading, 1989.

[10] Guerrero Víctor M. Guzmán. *Análisis Estadístico de Series de Tiempo Económicas*. Thomson Editores. México DF. 2003.

[11] Hansen James V. McDonald James B., Nelson Ray D. "Time Series Prediction with Genetic Algorithms designed neural networks: An empirical comparison with modern statistical models". *Computational Intelligence*, Vol. 15. No. 3. 1999 Pp.171-184.

[12] Holland, John H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbor 1975.

[13] Makridakis Spyros, Wheelwright Steven C. *Manual de Técnicas de Pronósticos*. Ed. Limusa. Mexico, DF. 1997.

[14] Makridakis Spyros, Wheelwright Steven C, Hyndman Rob J. *Forecasting: Methods and Applications*. John Wiley & Sons. New York. USA. Third edition 1998.

[15] Medeiros C. Marcelo, Veiga Álvaro. "A Hybrid Linear-Model for Time Series Forecasting". *IEEE Transactions on Neural Networks*, Vol. 11, No. 6, November 2000. Pp.1402-1412.

[16] Medeiros C. Marcelo, Veiga Álvaro. "A Flexible Coefficient Smooth Transition Time Series Model". *IEEE Transactions on Neural Networks*, Vol. 16, No. 1, January 2005. Pp. 97-113.

[17] Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer. Berlin. 1994

[18] Miller Brad L., Goldberg David E. "Genetic Algorithms, Tournament Selection, and the Effects of Noise". *Complex Systems*. Vol. 9. 1995. Pp. 193-212

[19] P. Flores, R. Garduño, L. B. Morales, and M. Valdez. "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*. Cuba 1999 Pp. 186-190.

[20] Sanchez-Corrales V.M. Flores-Perez Pedro, J.A. Valera. Perez-Tello M." Mass Balance Calculations in Copper Flash Smelting by Means of Genetic Algorithms". *Journal of Metals*. Vol. 56, No.12, 29-32, 2004.

[21] Szpiro George G. "Forecasting chaotic time series with genetic algorithms". *Physical Review E*. Vol. 5. No. 3. March 1997.Pp. 2557-2568.

[22] Yadavalli Vansi K., Dahule Rahul K., Tambee Sanjeev S., Kulkarni B. D. "Obtaining functional form for chaotic time series using genetic algorithm". *CHAOS* Vol. 9 No. 3. 1999. Pp.789-794