In: Proceedings of XXV Iberian Latin American Congress on Computational Methods in Engineering (CILAMCE), Recife (Brazil), 10-12/november, 2004. [CD-ROM]

A GENE EXPRESSION PROGRAMMING SYSTEM FOR TIME SERIES MODELING

Heitor S. Lopes
Wagner R. Weinert
hslopes@cpgei.cefetpr.br
weinert@brturbo.com
CPGEI – Centro Federal de Educação Tecnológica do Paraná
Av. 7 de setembro, 3165, 80230-910, Curitiba - PR – Brazil

Abstract. *This paper presents a heuristic method for time series modeling. The method is based on gene expression programming, a recently proposed evolutionary computation technique. We explain in details the method and the system we developed, named EGIPSYS. We consider time series modeling as a particular problem of symbolic regression. Five different series found in the literature were modeled, including river flows, sunspots and financial data. Performance was measured using the correlation coefficient, the coefficient of variation and the normalized mean square error. For each data series, a training set was used to find a suitable model, and an evaluation set was used to access its forecast performance. Results show that the heuristic method here proposed is appropriate for modeling time series and, also, that it can display a fair performance for one-step and multi-step forecasting.*

Keywords: *Time series, Forecast, Gene expression programming, Evolutionary Computation*

1. INTRODUCTION

Time series are a sequence of data points, measured typically at uniform time intervals. The analysis of time series may include many statistical methods that aim to understand such data by constructing a model. Models, in turn, are used to understand the underlying behavior of a dynamical physical system of process. They are intended to capture the significant features and represent them usually as mathematical equations. Using an appropriate model, one can predict (forecast) future events, based on the past ones. Therefore, time series modeling and time series prediction are two faces of the same coin. Sometimes, the time series modeling can be understood as an identification problem or, more generally, a symbolic regression problem.

As mentioned before, there are many statistical-based methods for modeling and forecasting time series and they are based on strong mathematical background. Most of these methods are based on ARMA-derived (AutoRegressive-Moving Average) methods. Many times, the complexity underlying these statistical methods precludes its use by those less acquainted with them. In the other hand, it is possible to use heuristic methods that give approximated, but satisfactory, solutions for such class of problems. This paper addresses this issue; we use a recently proposed evolutionary algorithm, namely, gene expression programming, for finding approximate models for time series. We show that using a specially built tool (EGIPSYS), it is possible to obtain simple mathematical models describing the temporal behavior of data, without deep knowledge about statistical methods.

This paper is structured as follows: first, we present the time series modeling problem more formally, and we cite some recent work on heuristic methods for this purpose. Next, we present the gene expression programming paradigm and the tool EGIPSYS. Then, the results of some computational experiments are reported. Finally, we present our conclusions and future directions of research.

2. TIME SERIES MODELING

Let a univariate dynamical system be represented by samples taken at a constant rate. The embedding theorem of Takens (1981) asserts that if the dynamical system is deterministic, the observed time series representing the system can be as in Eq. 1, where τ is the (constant) time-delay between samples, d is the embedded dimension and f is a function.

$$x(t) = f[x(t-\tau), x(t-2.\tau), \cdots, x(t-(d-1).\tau)] \tag{1}$$

The time series modeling problem can be formulated as follows: given n values of an observed series, find the appropriate d, τ and f. In words, the objective is to find a suitable mathematical model that can roughly explain the behavior of the dynamical system. There are statistical methods to cope with part of this problem, but they require a large amount of data, what is rarely the case in practical problems. Therefore, many heuristic methods have been employed.

The function *f(.)* is the "center of the storm", but, unfortunately, one does not have a full insight of the dynamical system to describe *f(.)* satisfactory by a simple equation or a set of equations with a finite number of parameters. Therefore *f(.)* might be defined by a complicated formulae or by functions not well known. Hence, this is a ill-posed problem since there could be many possible approximations to *f(.)* for the points given, with an arbitrary acceptable error. Again, heuristic methods emerge as an interesting alternative. This is an important point to recognize since most time one does not know anything about the time series being analyzed, including other data that may affect the physical system.

In general, to find a model to describe a time series, one uses all available data or part of it, leaving some for testing the model. Once finding a suitable model, it is possible to predict future events. This prediction can be one-step-ahead, for short-term prediction, or multi-step-ahead, for long-term prediction. It is something obvious that long-term predictions are subject to larger errors than short-term predictions. This is especially true when the series is not stationary, that is, some parameter of the underlying physical model is changes throughout time. Other issue that makes difficult obtaining a suitable model is the random noise inherent to the physical system or related to the measurement procedure.

2.1 Related work on evolutionary computation for time series modeling

Evolutionary computation models have been used for time series modeling in the past, mainly for chaotic, nonlinear and empirical time series. For instance, Oakley (1994) and Lee (2001) have used used genetic programming for modeling chaotic time series. Koza (1992), Kaboudan (1999) and Santini & Tettamanzi (2001), among others, have used genetic programming for modeling financial time series. Fogel and Fogel (1996) have used evolutionary programming to discriminate between chaotic signals and noise. Genetic programming was also used by Rodriguez-Vazquez (2001) and by Howard and Roberts (2002) for modeling, respectively, traffic data and meteorological data.

Time series prediction can be considered a particular case of a symbolic regression problem. Therefore, several researchers have suggested genetic programming for solving this class of problem, such as Koza (1992) and Augusto & Barbosa (2000). Recently, Ferreira (2003) have used a gene-expression programming approach for predicting time series.

3. GENE EXPRESSION PROGRAMMING

Gene expression programming (GEP) is a population-based evolutionary algorithm developed by Ferreira (2001) and it is a direct descendent of genetic programming (Koza, 1992). In GEP, individuals are encoded as linear strings of fixed size (genome) such as in Fig.1, which are expressed later as non-linear entities with different size and shapes. These entities are known as expression trees (ETs). Usually, individuals are composed by only one chromosome, which, in turn, can have one or more genes, divided in head and tail parts. ETs are the expression of a chromosome, and they undergo the selection procedure (usually fitness proportionate), guided by their fitness value, so as to generate new individuals. During reproduction, the chromosomes, rather than the respective ET, are modified by the genetic operators.

In GEP, there are several genetic operators specially devised for this kind of representation: replication, mutation, IS transposition, RIS transposition, root transposition, gene transposition, single or double recombination (crossover), and gene recombination. A detailed description of these operators can be found in Ferreira (2001).

Like other evolutionary algorithms, GEP also starts with a randomly-generated population of solutions. Next, each individual of this population is evaluated: chromosomes are expressed as ETs and submitted to a fitness function that measures the goodness of the corresponding solution. A stop criterion is checked (for instance, quality of the best solution, timing-out, etc) and, if it is met, the evolutionary process finishes. Otherwise, the individuals of the population are submitted to a selection procedure which implements the Darwinian survival-of-the-fittest rule. Highly-fitted individuals are probabilistically selected and, later, they are changed by means of the genetic operators previously cited. The best individual of a generation is always kept for the next. Hopefully, the application of the genetic operators can improve quality of the individuals and allow the algorithm to better explore the solution's

search space. This new generation is evaluated and the process is repeated until the stop criterion is met.

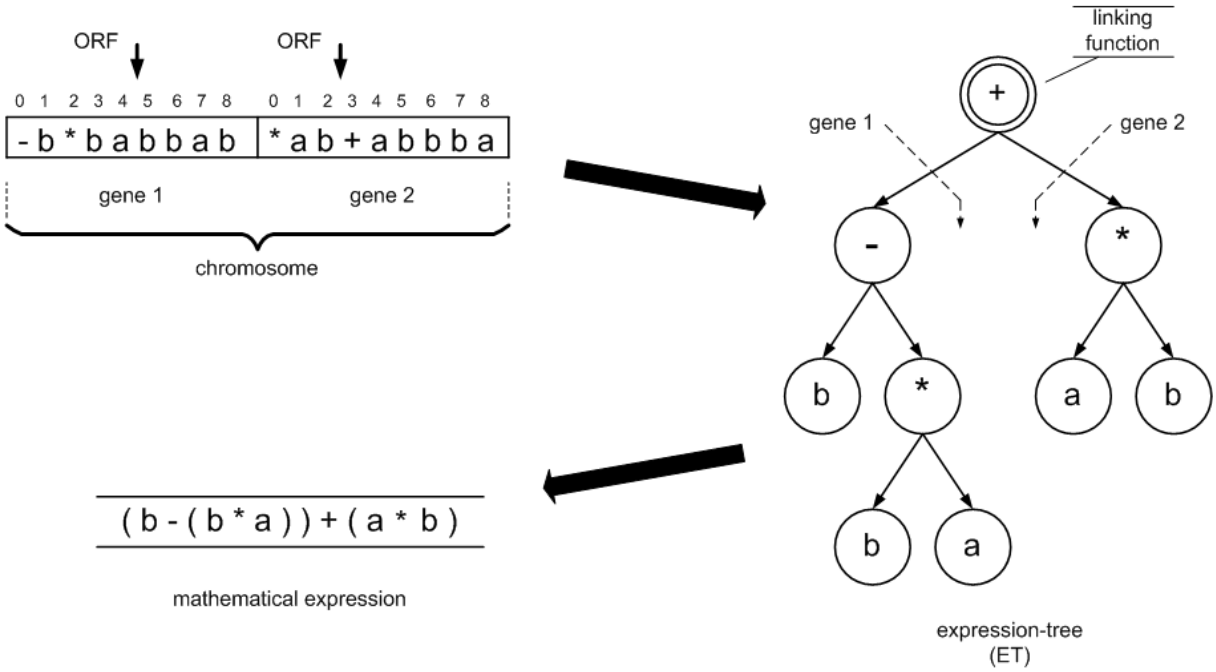

Figure 1 – Individual representation in GEP: two-genes chromosome, expression-tree and corresponding mathematical expression.

3.1 EGIPSYS

EGIPSYS (*Enhanced Gene-expressIon for Symbolic regression problemS*) is a GEP-based system developed by Lopes and Weinert (2004) and includes several improvements, making it a powerful and easy-to-use tool for symbolic regression problems.

Differently from GEP, EGIPSYS can work with variable-length chromosomes. For the generation of the initial random population, the possible range of gene head sizes is limited by a user-defined parameter. Then, 50% of the population is randomly generated with gene head size ranging from one to the user-defined parameter; the rest 50% is generated uniformly with all possible gene head sizes from one to the parameter. The use of variable-length chromosomes allows a larger genetic diversity throughout generations.

EGIPSYS also defines a new policy for handling constants as terminal nodes of the trees. User simply defines a probability for using constants (within candidate solutions), quite differently from the complicate strategies devised by Ferreira (2001). Besides, user can define the initial range of these constants.

GEP originally uses roulette-wheel as selection method. EGIPSYS implements this method and also the stochastic tournament selection method, which imposes less selective pressure, allowing a smoother evolution.

The genetic operators defined by Ferreira (2001) were implemented in EGIPSYS as originally defined, except that they were adapted to work with variable-length chromosomes. Besides, we implemented a new local-search operator aimed at fine-tuning constants. Since this operator is computational intensive, it is applied with parsimony by means of a user-defined probability.

In EGIPSYS, several built-in fitness functions were defined. All these functions yield a normalized value between 0 (the worst) and 1 (the best), and they are based either on the mean absolute error, the mean square error or the Pearson's correlation coefficient. Together

with the selection method, user can set on or off an auxiliary mechanism to control the selective pressure by using a linear fitness scaling (similar to that used in genetic algorithms). This mechanism always keeps the average fitness value of the population and compress or decompresses the scale to avoid large differences between individuals.

EGIPSYS has a graphical user interface that allows easy interaction for setting parameters. It also can generate an on-line evolution graphics (with minimum, average and maximum fitness of the population in all generations) and allows user to stop running, see partial (best) results and resume running without losing stored values. The system also generates a graph showing given data points and the points obtained by the application of the best individual to the fitness cases. The system allows batch execution and generates several control files for statistical analysis. EGIPSYS was designed for easy interaction with the user and can be applied to many problems without requiring a deep knowledge of the underlying evolutionary process. This software will be put in public domain as an aid to foster more research in this area (*http://bioinfo.cpgei.cefetpr.br/en/softwares/index.html*).

3. EXPERIMENTS AND RESULTS

3.1 Methodology

Data used in this work are from a time-series data repository (*http://www.stats.uwo.ca/faculty/aim/epubs/mhsets/)* created by Hipel & McLeod (1994).

For all results, the mathematical expression presented uses T_i to represent the x_{t-i} term. The final model was obtained by simple algebraic simplification.

There are many ways to evaluate performance of the obtained models, but no consensus on this issue. In this work, models were evaluated using three measures: the well-known Pearson's correlation coefficient (R), defined by Eq. (1); the coefficient of variation (CV), defined by Eq. (2); and the normalized mean square error (NMSE), defined by Eq. (3).

The correlation coefficient measures the suitability of the adjusted model to the data, ranging from -1 to +1. A positive value means a positive linear correlation; a negative value, the opposite. Values of R close to zero mean bad or no correlation between data and the adjusted model. CV measures the relative scatter in data with respect to the mean and, therefore, the smaller, the better. This measure was already used by Iba et al. (1994) and Lee (2001) for time-series analysis. The NMSE is a method to compare the mean of a series against the predicted values and it is frequently used in the analysis of time series (Gupta et al, 1998). If the NMSE is greater than the unity, then the predictions are doing worse than the series mean. In the other hand, if the NMSE is less than the unity, then the forecasts are doing better than the series mean. In equations (1), (2) and (3), $\bar{x}$, x_i , $\tilde{x}_i$ and σ^2 stand, respectively, for the sample average of observations, the real *i*-th observation, the corresponding *i*-th value given by the model and the variance of the sample.

$$R = \frac{N \cdot \sum_{i=1}^{N} (x_i \tilde{x}_i) - \left(\sum_{i=1}^{N} x_i \right) \cdot \left(\sum_{i=1}^{N} \tilde{x}_i \right)}{\sqrt{\left[N \cdot \sum_{i=1}^{N} (x_i)^2 - \left(\sum_{i=1}^{N} x_i \right)^2 \right] \cdot \left[N \cdot \sum_{i=1}^{N} (\tilde{x}_i)^2 - \left(\sum_{i=1}^{N} \tilde{x}_i \right)^2 \right]}} \tag{1}$$

$$CV = \frac{1}{\bar{x}} \sqrt{\left[\frac{1}{N} \cdot \sum_{i=1}^{N} (x_i - \tilde{x}_i)^2 \right]} \tag{2}$$

$$NMSE = \frac{1}{\sigma^2}\left[\frac{1}{N}\cdot\sum_{i=1}^{N}(x_i - \tilde{x}_i)^2\right] \quad (3)$$

In the experiments shown here, the default parameters used in EGIPSYS were somewhat different from those defined in Lopes & Weinert (2004). We used the following parameters: number of genes: 5; initial range for gene head: 6-10; linking function: sum; function set: {+,-,*,%} (except when explicitly mentioned); terminal set: samples corresponding to the embedded dimension, unit constant (1) and random constants (in the initial range -10..+10); selection method: tournament selection (with 7% of the population size); automatic fitness scaling: yes; operators and probabilities: cloning, replication, mutation (0.25), IS, RIS and gene transpositions (0.15), recombination (0.95), gene recombination (0.15), local search operator (0.15). The local search operator was activated every (number of generations/10) generation. The fitness function was the standard one proposed in EGIPSYS and was based on the sum of the absolute errors. The normalization factors of the fitness function (see Lopes & Weinert (2004) for details) as well as the number of individuals and generations were set differently for each of the analyzed time series.

In this paper, we addressed both short-term prediction and long-term prediction using the models obtained by EGIPSYS. The methodology used was almost the same for both cases, except that for short-term prediction all but a small number of data were used for creating the model (the rest was used for evaluating it), and for long-term prediction, only half of the data was used to create the model, and the rest was used for evaluation.

3.2 Furnas series

This data series represents the monthly unregulated Rio Grande river flow at Furnas dam, in Brazil, given in cubic meters/second, from 1931 to 1978. The series has 576 points and was introduced by Noakes et al. (1985).

Considering a yearly basis for the seasonality, the embedded dimension was set to 12 and time-delay to 1. We used 564 points for training and 12 to compare actual and forecasted values. After, we extended the prevision to the next 12 months-period. The best model presented found by EGIPSYS had 45 nodes and was obtained in 500 generations using 50 individuals. The other parameters were set to the standard values presented before. Fig. 2 shows the mathematical expression obtained and Eq. (4) its algebraic simplification. For this series, we obtained R=0.8158, CV=0.3896 and NMSE=0.3445 with the training set; and R=0.9301, CV=0.2058 and NMSE=0.0853 with the evaluation set. Fig. 3 shows the original data, the curve obtained with the model and the multi-step-ahead forecast.

```
(?11.500/(T4/((T11*T10)/T4)))|+|((T1+((T11-T9)/T4))+?0.000)|+|((((T1-T9)/?-
1.360)-T3)/?7.353)|+|(?11.500/(T4/((T11*T10)/T4)))|+|(?15.001/(T1/T3))
```

Figure 2 - Mathematical expression found for the Furnas series.

$$x_t = \left(\frac{23.\left(x_{t-11}.x_{t-10}/x_{t-4}\right)+x_{t-1}.x_{t-4}+x_{t-11}-x_{t-9}}{x_{t-4}}\right) - 0.1.\left(x_{t-1}-x_{t-9}+1.36.x_{t-3}\right)+\left(\frac{15.x_{t-3}}{x_{t-1}}\right) \quad (4)$$

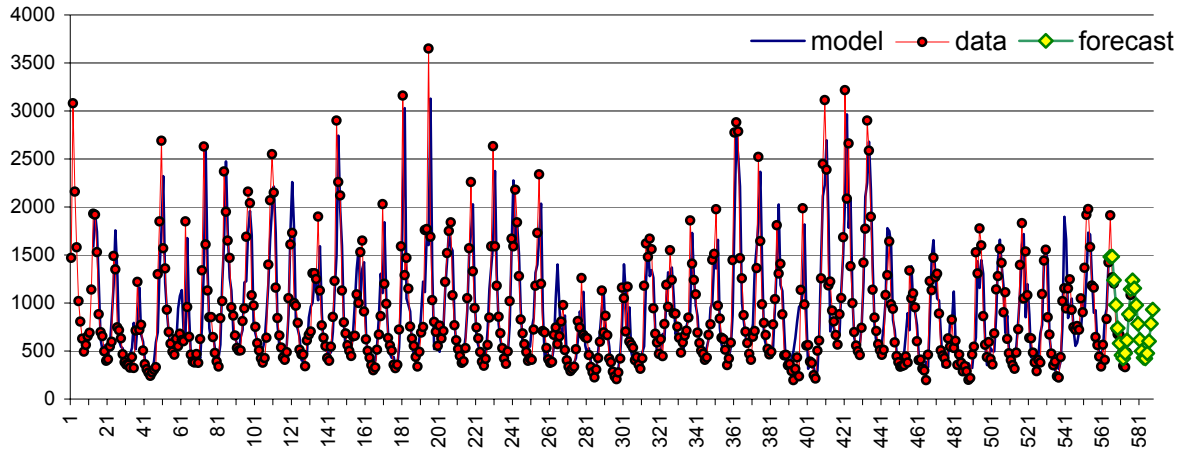

Figure 3 - Results for the Furnas series.

3.3 Nile river series

This series represents the annual minimum level of Nile River (Egypt), from year 622 to 1921. Apparently, Hipel & McLeod (1994) have introduced this time series and there is no further information about it. The series has 1297 data points (3 of them are missing) and we have not considered any a priori seasonality on the data. Therefore, the embedded dimension was set arbitrarily to 5, and the time-delay to 1. We used 1292 points for training and 5 to compare actual and forecasted values. After, we extended the prevision to the next 5 periods.

The best model found by EGIPSYS had 43 nodes and was obtained in 200 generations using 30 individuals. For this series, we obtained R=0.7726, CV=0.0611 and NMSE=0.4294 with the training set; and R=0.3468, CV=0.0227 and NMSE=0.0964 with the evaluation set. Fig. 4 shows the mathematical expression obtained and Eq. (5) its simplification. Fig. 5 shows the data above observation 1000 (to avoid an excessive number of points in the graph), including the original data, the curve obtained with the model and some multi-step-ahead forecast.

```
-T1T3-T3?9.766T5/+?4.483T5T1T1T5T3T4T3T5T1T3?-2.60/-T3T5T1*-?0.661T3T3?-
9.13?4.282T2?6.018T2T2?-1.77T3T2T2T1-T3//T3*-?0.594T1T3T3T5T4?5.908T5?3.060
?-2.17?-4.53?6.306T3T2/-T4T4T1?-1.79*-+?-9.16T4T5T1?-2.59T5T5?-1.79T2T4T1
T4//T1/-*/*?2.824?-9.58T4T5-5.89T1?6.729T3T3?-?1.41T4T3?-2.34
```

Figure 4- Mathematical expression found for the Nile river series.

$$x_t = 1 + x_{t-1} - x_{t-3} - \frac{x_{t-1}}{x_{t-4}} + \left(\frac{x_{t-3}^2 - x_{t-1} + x_{t-5} - 1}{x_{t-3}} \right) - \left(\frac{(56.426 x_{t-4}) / x_{t-5}}{(6.729 x_{t-1} - 2.824) \cdot x_{t-1}} \right) \tag{5}$$

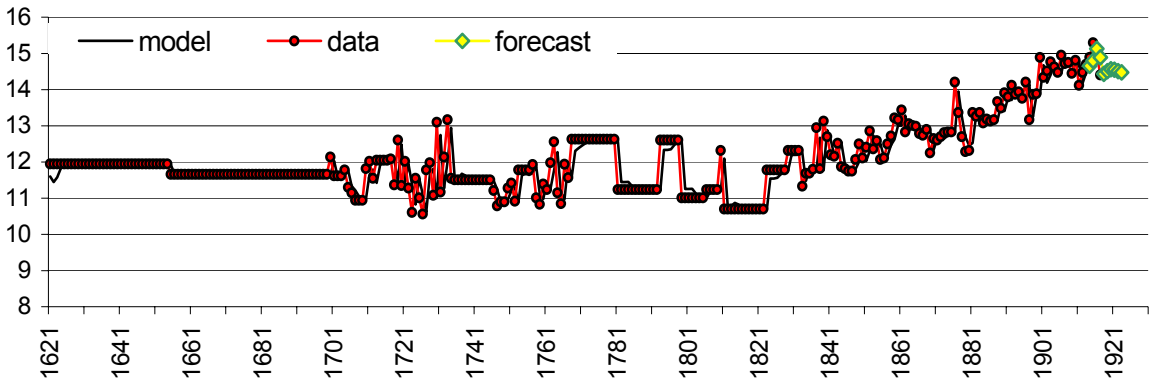

Figure 5 - Results for the Nile river series.

3.4 Tietê river series

Similar to those series before, this is also related to a hydrological problem. This data set is about the monthly outflow of Tietê river, at Cumbica station, in Brazil, from 1948 to 1978. For this sort of data, it is expected a yearly basis of seasonality and, therefore, the embedded dimensions was set to 12 (time-delay was set 1). This series has 372 data points, and we used 360 points for training and 12 to compare actual and forecasted values. After, as an exercise of multi-step-ahead forecast, we extended the prevision to the next period of 12 months.

Experiments using the previously mentioned parameters for EGIPSYS did not succeed satisfactory results and, for this particular data set, we used 300 individuals during 100 generations. Also, we included the sine function in the function set and constant π in the terminal set. The best model found had 72 nodes and we obtained R=0.7697, CV=0.4093 and NMSE=0.4160 with the training set; and R= 0.8586, CV=0.2149 and NMSE=0.0975 with the evaluation set. Fig. 6 shows the resulting mathematical expression corresponding to Eq. (6). Fig. 7 shows the original data, the curve obtained with the model and the multi-step-ahead forecast.

```
(((((T11-T1)-(?0.000+T1))/?0.889)+(sin(T3*?0.074)))/(?0.403*T9))+(sin((T12*
T1)/((sin?3.455)/T9)))+(((T10/T4)-(sin((T9/?0.000)*(T1+T8))))+T1)+(sin(T12-
(?1.034*((T10/((sinT8)*T9))*?0.894))))+(T11/(T6*(?0.557-((sinT5)/((T7-T9)*
?3.139)))))
```

Figure 6 - Mathematical expression found for the Tietê river series.

$$x_t = \frac{x_{t-11} + 0.889 \cdot sin(0.074 \cdot x_{t-3})}{0.3583 \cdot x_{t-9}} + sin\left(\frac{x_{t-9} \cdot x_{t-12} \cdot x_{t-1}}{-0.3083}\right) + \frac{x_{t-10}}{x_{t-4}} - sin\left(\frac{x_{t-1}}{x_{t-8}}\right) + \cdots$$
$$\cdots + x_{t-1} + sin\left(x_{t-12} - \frac{0.9244 \cdot x_{t-10}}{x_{t-9} \cdot sin(x_{t-8})}\right) + \frac{x_{t-11}}{x_{t-6} \cdot \left(0.557 - \frac{0.3186 \cdot sin(x_{t-5})}{x_{t-7} - x_{t-9}}\right)} \quad (6)$$

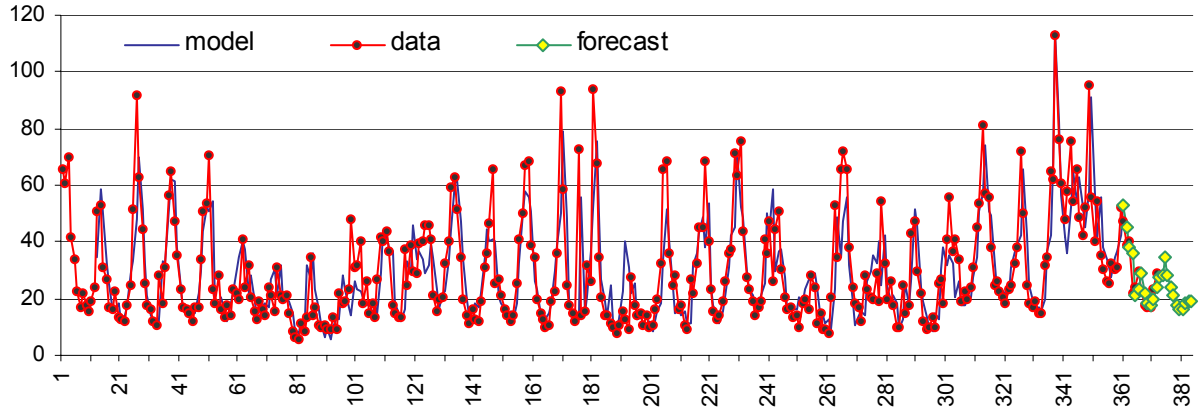

Figure 7 - Results for the Tietê river series.

3.5 Sunspots series

This data set is related to the annual number of sunspots, registered from year 1700 to 1988. Possibly, this series was introduced by Weigened et al. (1992) and, since then, it has been widely used in the time series literature. This series was also used by Ferreira (2003) for a similar experiment using GEP. In most experiments, only the 100 observations (observation 71 to 171), out of the 289, are used. We followed the same approach and, arbitrarily, we defined the embedded dimension to 10. We used the next 10 observations (from 1800 to 1809) for evaluating the forecast, and after we extended the prevision to the next period of 10 points.

For this run, we used 100 individuals during 100 generations and the best model found had 23 nodes. We obtained R=0.8151, CV=0.4794 and NMSE=0.3581 with the training set, and R=0.9190, CV=0.3693 and NMSE=0.1422 with the evaluation set. Fig. 8 shows the mathematical expression corresponding to Eq. (7). As can be seen, Eq. (7) is a very simple model. This is an evidence of the parsimony capability of EGIPSYS. Fig. 9 shows the original data, the curve obtained with the model and some multi-step-ahead forecast.

```
((((T7+T7)+T1)/(T3/T2))/T2)+((?-2.85-(T3/?6.647))-T10)+(T1+T10)
```

Figure 8 - Mathematical expression found for the Sunspots series.

$$x_t = \frac{2 \cdot x_{t-7} + x_{t-1}}{x_{t-3}} - \frac{18.944 + x_{t-3}}{6.647} + x_{t-1} \tag{7}$$

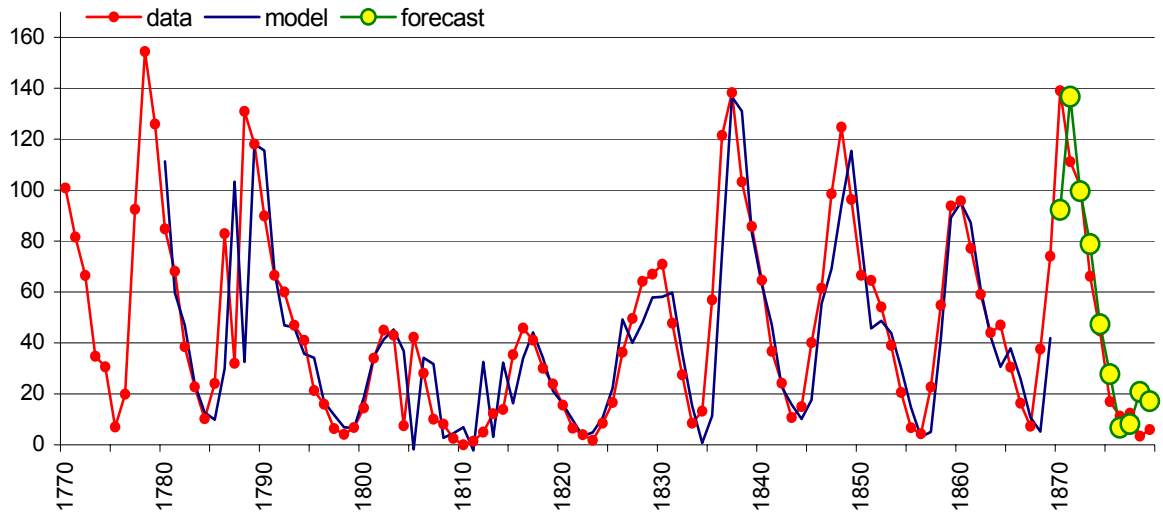

Figure 9 - Results for the sunspots series.

3.6 Daily wages series

This data is related to the real daily wages in pounds in England, from year 1260 to 1984. There is no reference of the first use of this data, except Hipel & McLeod (1994). There are 735 observations and we used 725 points for training and 10 to compare actual and forecasted values. The embedded dimension was set to 10 and time-delay to 1. Unlike the previous experiments, we did not extend the prevision beyond the last observation because this series seemed much easier than the previous to be modeled.

The best model found by EGIPSYS had only 21 nodes and it is shown in Fig. 10, corresponding to Eq. (8). We obtained R=0.9957, CV=0.0811, NMSE=0.0095 for the training set; and R=0.9806, CV=0.006 and NMSE=0.0414 for the evaluation set. Fig. 11 shows the original data, the curve obtained with the model and some multi-step-ahead forecast.

```
(T7/T2)+((T5/((T7+T8)+?0.579))/((T5+?0.996)/T4))+(?-1.37+T1)
```

Figure 10 - Mathematical expression found for the daily wages series.

$$x_t = \frac{x_{t-7}}{x_{t-2}} - \frac{x_{t-5}}{x_{t-7} + x_{t-8} + 0.579} \cdot \frac{x_{t-4}}{x_{t-5} + 0.996} + x_{t-1} - 1.37 \tag{8}$$

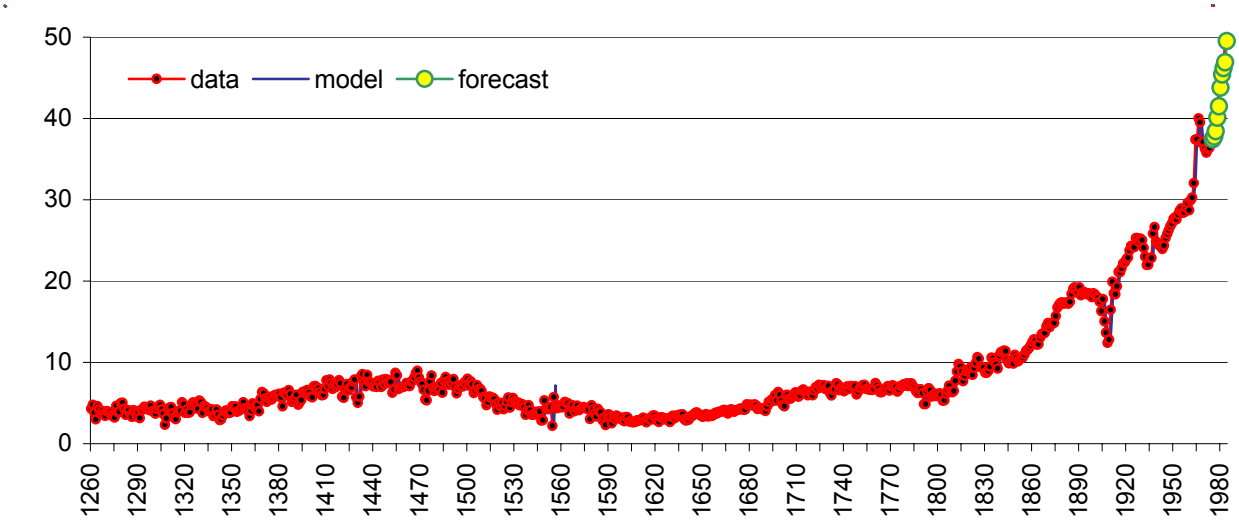

Figure 11- Results for the daily wages series.

3.7 One-step ahead forecast

Based on the models found for the problems presented before, we calculated the specific error for the one-step-ahead forecast. This procedure is usual in most forecast methods, since multi-step-ahead forecast is much more difficult and subject to major errors. Table 1 summarizes these results.

Table 1. Error for one-step-ahead forecast

Series	One-step-ahead forecast error
Furnas	-15.29 %
Nile	-1.51 %
Tietê	+13.32 %
Sunspots	-7.76 %
Daily wages	-0.61 %

4. DISCUSSION AND CONCLUSIONS

We have developed a gene-expression programming system – EGIPSYS – specially devised for symbolic regression problems. In this work, it was shown its utility for time-series modeling problems.

In this paper we used several river flow data sets. It is well known that modeling this sort of data is frequently a challenging exercise, since the univariate time series does not explicitly include physical processes involved in the hydrological cycle, such as meteorological data. The other time series, sunspots and daily wages, are more well-behaviored and, as expected, results were much better.

For all time series, the R was from 0.7726 (Tietê) to 0.9957 (daily wages) using the training set, indicating that the models found have followed reasonably the trend in raw data. The CV was somewhat high for all series but those whose absolute values were small (namely, Nile and daily wages). This behavior was consistent in both training and evaluation sets. The NMSE has no important meaning for the training set. Overall, the analysis of results

using the training sets suggests the appropriateness of the models. This is clearly seen in the graphics presented and an important achievement of the method.

The analysis of results over the evaluation sets shows how good the models found do regarding the forecast. As shown in Table 1, the one-step-ahead forecast error was largely ranged. Both Furnas and Tietê series were the most difficult to deal with, given the number of nodes of the model found. Although they could capture well the behavior of data, the models can be somewhat overfitted, leading to high individual errors. The small R for the Nile river series (0.3468) is anomalous and, possibly, it is due to the small number of observations computed (only 5). In the other hand, for all other series, R was much higher than those for the corresponding training set. In the same way, for all series, both CV and mainly NMSE were small, clearly indicating a satisfactory forecast.

It must be kept in mind that in time series analysis, one may have two aims: either to understand the physical system (from which the time series was generated) or to predict future events, based on the past ones. A given method can be suitable for the former task but perform badly for the later. This paper has presented a heuristic method primarily aimed to model time series, but we have shown also its utility for forecasting.

It is a matter of fact that the method presented here is not able to capture the full behavior of the underlying dynamical system represented by the time series. In special, the small high-frequency transitions (under the Fourier spectral analysis) are quite difficult to be modeled. This is particularly true not only for this method but also for most time series modeling methods. In the other hand, the analysis of time series using heuristic methods, in general, avoids the burden of complex mathematical methods that require a deep statistical analysis (such as ARMA-derived methods), deseasonalizing or previous transformations of raw data.

Regarding the complexity of the models found, it was already shown elsewhere (Lopes & Weinert, 2004) that the method can find solutions quite smaller than those found by regular genetic programming systems, thanks to the encoding principle of GEP and the enhanced characteristics of EGIPSYS. The number of nodes of the best solutions found, as a direct measure of the hardness in modeling the time series, suggests that the method can successfully control the bloat effect (code growth) and, therefore, provides parsimonious solutions.

Acknowledgements

Authors would like to thank CAPES for the MSc grant to W.R. Weinert and CNPQ for the research grant to H.S. Lopes (process 350053/2003-0).

REFERENCES

Augusto, D.A. & Barbosa, H.J.C., 2000. Symbolic regression via genetic programming. In: *Proceedings of Sixth Brazilian Symposium on Neural Networks*, pp.173-178. Piscataway: IEEE Press.

Ferreira, C., 2001. Gene expression programming: a new adaptive algorithm for solving problems. *Complex Systems*, vol. 13, n. 2, pp. 87-129.

Ferreira, C., 2003. Function finding and the creation of numerical constants in gene expression programming. In: Benitez, J.M., Cordon, O., Hoffmann, F. & Roy, R., eds, *Advances in Soft Computing: Engineering Design and Manufacturing*, pp. 257-266. Berlin: Springer-Verlag..

Fogel, D.B. & Fogel, L.J., 1996. Preliminary experiments on discriminating between chaotic signals and noise using evolutionary programming. In: Koza, J.R et al, eds, *Proceedings of 1996 Conference on Genetic Programming,* pp. 512-520. Cambridge: MIT Press.

Gupta, A., Bansal, K. & Vadhavkar, S., 1998. Neural networks based forecasting techniques for inventory control applications. *Data Mining and Knowledge Discovery*, vol. 2, pp. 97-102.

Hipel, K.H. & McLeod, A.I., 1994. *Time Series Modeling of Water Resources and Environmental Systems*. New York: Elsevier.

Howard, D. & Roberts, S.C., 2002. Application of genetic programming to motorway traffic modelling. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1097-1104. San Francisco: Morgan Kaufmann.

Iba, H., de Garis, H. & Sato, T., 1994. Genetic programming using a minimum description length principle. In: Kinnear Jr., ed, *Advances in Genetic Programming*, pp. 265-284. Cambridge: MIT Press.

Kaboudan, M., 1999. A measure of time series predictability using genetic programming applied to stock returns. *Journal of Forecasting*, vol. 18, pp. 345-357.

Koza, J.R., 1992. *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. Cambridge: MIT Press.

Lee, G.Y., 2001. Time series perturbation by genetic programming. In: *Proceedings of the 2001 Congress on Evolutionary Computation*, pp. 403-409. Piscataway: IEEE Press.

Lopes, H.S. & Weinert, W.R., 2004. EGIPSYS, a enhanced gene-expression programming system for symbolic regression problems. To appear in *International Journal of Applied Mathematics and Computer Science*, vol. 14.

Noakes, D.J., McLeod, A.I. & Hipel, K.H., 1985. Forecasting monthly riverflow time series. *International Journal of Forecasting*, vol. 1, pp. 179-190.

Oakley, H., 1994. Two scientific applications of genetic programming: stack filters and non-linear equation fitting to chaotic data. In: Kinnear Jr., ed, *Advances in Genetic Programming*, pp. 369-389. Cambridge: MIT Press.

Rodriguez-Vazques, K., 2001. Genetic programming in time series modeling: an application to meteorological data. In: *Proceedings of the 2001 Congress on Evolutionary Computation*, pp. 261-266. Piscataway: IEEE Press.

Santini, M. & Tattamanzi, A., 2001. Genetic programming for financial time series prediction. In: *Genetic Programming, Proceedings of EuroGP'2001*, *LNCS*, vol. 2038, pp. 361-370. Berlin: Springer-Verlag.

Takens, F., 1981. Detecting strange attractors in turbulence. In: Hand, D., Young, L.S., eds, *Dynamical Systems and Turbulence*, pp. 366. Berlin: Springer-Verlag.

Weigened, A.S., Huberman, B.S. & Rumelhart, D.E., 1992. Predicting sunspots and exchange rates with connectionist networks. In: Casdagli, M. & Eubank, S., eds, *Nonlinear Modeling and Forecasting*, pp. 395-432. Boston: Addison-Wesley.