An. Şt. Univ. Ovidius Constanţa Vol. 17(3), 2009, 45–68

ALTERNATIVE MODELS IN PRECIPITATION ANALYSIS

Alina Barbulescu and Elena Bautu

Abstract

Precipitation time series intrinsically contain important information concerning climate variability and change. Well-fit models of such time series can shed light upon past weather related phenomena and can help to explain future events. The objective of this study is to investigate the application of some conceptually different methods to construct models for large hydrological time series. We perform a thorough statistical analysis of the time series, which covers the identification of the change points in the time series. Then, the subseries delimited by the change points are modeled with classical Box-Jenkins methods to construct ARIMA models and with a computational intelligence technique, gene expression programming, which produces non-linear symbolic models of the series. The combination of statistical techniques with computational intelligence methods, such as gene expression programming, for modeling time series, offers increased accuracy of the models obtained. This affirmation is illustrated with examples.

1 Introduction

Time series analysis and prediction is a problem of high importance in almost all fields of modern science. Regardless of their nature (e.g. temperature

Key Words: ARIMA, Gene Expression Programming, Modeling, Precipitation, Statistical analysis, Time series

Mathematics Subject Classification: 65C60, 46N30, 62H11, 03F60.

Received: April 2009

Accepted: October 2009

This work was supported by UEFISCSU, Romania, under Grant ID_262/2007 and Grant TOMIS (11-041/14.09.2007)

records, series of stock prices etc.) time series that monitor real world phenomena are hard to summarize and more importantly, difficult to predict. Precipitation time series contain information that may help to detect and explain issues concerning climate variability and change. Well fit models of such time series are used to analyze past weather related phenomena and can shed light upon future events.

We study the series of mean annual precipitation records and mean monthly precipitation records registered in the period 1965 - 2005 at Medgidia meteorological station, situated in Dobrudja region, in the South - East of Romania. Such series are gathered from dynamic environments, therefore are influenced by many factors. The process describing the data may undergo changes during its evolution that trigger the appearance of points in the time series where the behavior of the series gathered changes [9].

In this study, we consider the problem of modeling time series by splitting it into blocks that belong to the same process, and constructing models for these blocks with classical and heuristic approaches. The goal is to find a good combination of the number of change points and models to fit the segments. Time series analysis is performed in a three steps methodology. First, the change point problem is addressed. Then, the models are derived, using both classical and modern methods. ARIMA models are constructed for the time series and the subseries delimited by the change points. An adaptive variant of the Gene Expression Programming (GEP) algorithm is also used for this purpose. Since the statistical methods used for change point detection gave contrasting results, we present the models obtained on the subseries delimited by change points, as well as for the entire time series. We also report some results obtained by combining an autoregressive model with GEP.

2 Related Work

Time series analysis methods may be broadly classified as classical and heuristic methods [14]. Classical methods include exponential smoothing, autoregressive or threshold methods [17], while heuristic approaches use mostly neural networks or evolutionary computation [14]. In usual approaches, the main assumption is that the process that generated the data is constant, therefore once a model that fits a given time series is obtained, it is used to analyze and predict it. Many time series that deal with real-world environments are affected by the permanently changing conditions, more or less abrupt, in the environment. The location where the data generating process of a time series changes is generally referred to as change point. In weather related time series, such shifts are known to occur [9].

Change point detection has been the focus of many studies in the statis-

tical literature. The works of Pettitt [15] and Buishard [4] are seminal in the literature. They propose statistical tests to find of a change point in the mean of a non-stationary time series, that leads to the time series being divided into two stationary time series. The disadvantage is that they test whether there is only one change point in the series. A common approach is to use sequential detection methods, coupled with piecewise regression analysis of the series. Lai [13] reviews sequential change point detection procedures. Hubert proposes a procedure that builds on the works of Klemes and Potter and optimally yields a partition of the series in many subseries [9]. Heuristic approaches for change point detection have also been reported. Jann [10] uses a genetic algorithm to identify multiple shifts in the course of a time series.

Once the change points are identified, the problem of identification of a suitable model is split into the determination of models that describe the subseries delimited by the change points. At this step, both classical and modern time series modeling methods may be used. Piecewise linear approximation (PLA) algorithms are highly used in the literature. In [19] authors report good results obtained in conjunction with a symbolic representation. PLA is used to model segments between feature points in [20]. Its major disadvantage is the approximation of the subseries by linear models. The work presented by Davis et. al in [5] is a hybrid approach, that combines metaheuristics with classical statistical methods, namely the autoregressive model (AR). They use a genetic algorithm for change point detection followed by autoregressive modeling of the segments determined. Sato et. al. [16] use wavelet expansions to find locally stationary autoregressive models. De Falco et al. constructed a genetic programming based system for forecasting time series and utilized it to perform predictions concerning El Nino forecast [7]. A different approach based also on genetic programming was used in [1], with an emphasis on the accuracy of the predictive solutions discovered by the algorithm.

3 Statistical analysis of the data

In the following, we shortly present the procedures and statistical tests used for analysis of the time series data.

A time series model for the observed data (x_t) is a specification of the joint distributions of a sequence of random variables (X_t) of which (x_t) is postulated to be a realization.

A. We study the normality of the time series by means of *Q–Q plot diagrams* and the *Jarque - Bera test* [18].

In Q–Q plot diagrams the data are plotted against a theoretical normal distribution in such a way that the points should form an approximate straight line. Departures from this straight line indicate departures from normality.

The null hypothesis of the Jarque - Bera test [17] is:

H_0: The process is normally distributed.

The test statistic JB is defined as

$$JB = \frac{n}{6}\left(S^2 + \frac{(K-3)^2}{4}\right),$$

where:

- n is the number of observations (or degrees of freedom in general);
- S is the sample skewness,
- K is the sample kurtosis.

The statistic JB has an asymptotic chi - square distribution with two degrees of freedom. If at a significance level α, $JB > \chi^2(2)$, then the hypothesis that the series is normally distributed is rejected.

B. The correlation of the time series is investigated analyzing *the autocorrelation function.*

The autocorrelation function at lag $h, (h \in \mathbf{N}^*)$, associated to a time series (X_t) [6] is defined by:

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}, \tag{1}$$

where $\gamma(h) = Cov(X_t, X_{t+h}),\ h \in \mathbf{N}^*$ is the autocovariance function of (X_t) at lag h.

If $x_1, x_2, \ldots, x_n$ are realizations of (X_t), by a formula analogous with (1), the empirical autocorrelation function, denoted by ACF, is obtained.

C. For the existence of change points, our investigation involves several statistical methods. For the Pettitt test and and the Buishard test, the null hypothesis is:

H_0: There is no change in the time series.

The Buishard test [4] works in the hypothesis that the series is normal. Since this is not always true for a real-world time series, we also use the Pettitt test [15], which is a nonparametric test and can be used even if the time series distribution is unknown.

As mentioned in previous section, classical statistical test, like the one described above, assume that only one change point exists in the data, which is a rather hard constraint. For this reason, in this study we also use CUSUM

charts [11] and the Hubert segmentation procedure [9] to detect whether multiple breaks in time series and the moments of their apparition. CUSUM charts are built by calculating and plotting the cumulative sum of differences between the values and the average. This tool reveals changes in the mean of the process generating the time series.

D. Finally, the homoscedasticity [17] is investigated with Barlett's test. Bartlett's test is used to test homogeneity of variances between k groups. The null hypothesis is:

$$H_0\colon\ \sigma_1 = \sigma_2 = \cdots = \sigma_k$$

and the alternative hypotheses is

$$H_1\colon\ \sigma_i \neq \sigma_j \text{ for at least one pair } (i, j).$$

The test statistic is

$$T = \frac{(n-k)\ln s_p^2 - \sum_{i=1}^{k}(n_i - 1)\ln s_i^2}{1 + \frac{1}{3(k-1)}\left(\sum_{i=1}^{k}\frac{1}{n_i - 1} - \frac{1}{n-k}\right)}$$

where

- n is the sample volume,
- k is the number of groups in which the sample is divided,
- n_i is the sample size of i^{th} group,
- s_i is the variance of the i^{th} group,

and

$$s_p^2 = \frac{\sum_{i=1}^{k}(n_i - 1)s_i^2}{n-k}.$$

The null hypotheses is rejected if $T > \chi^2_{\alpha,k-1}$, where $\chi^2_{\alpha,k-1}$ is the upper critical value of the chi-square distribution with $k-1$ degrees of freedom at the significance level of α.

Since this test is sensitive to departures from normality, it was applied after the transformation of the series that were not normal into normal series.

4 Time series modeling methods

The problem of modeling time series (or the segments bounded by the change points) is tackled with a classical approach using autoregressive processes and with a heuristic approach, based on gene expression programming and an adaptive variant of this algorithm. The methods used are briefly described in the following sections.

4.1 ARIMA models

For a detailed description of the Box-Jenkins methodology, we recommend [6]. In the following, a brief enumeration of the notions used later on in the paper is given.

Let (X_t) be a discrete process in time and let us consider the operators defined by:

$$B(X_t) = X_{t-1},$$

$$\Phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p,\ \phi_p \neq 0,$$

$$\Theta(B) = 1 - \theta_1 B - \cdots - \theta_q B^q,\ \theta_q \neq 0,$$

$$\Delta^d(X_t) = (1 - B)^d X_t.$$

B is called a backshift operator.

(X_t) is said to be an $ARIMA(p, d, q)$ process if $\Phi(B)\Delta^d(X_t) = \Theta(B)\epsilon_t$, where the absolute values of the roots of Φ and Θ are greater than 1 and (ϵ_t) is a white noise. Particular cases are:

$$ARMA(p, q) = ARIMA(p, 0, q),$$

$$AR(p) = ARIMA(p, 0, 0),$$

$$MA(q) = ARIMA(0, 0, q).$$

4.2 Evolutionary approach

The computational intelligence technique used to discover time series models pertains to the broad class of evolutionary computation techniques. One of the main research areas in the field of evolutionary computation is the Genetic Programming (GP) paradigm developed in early '90s by John Koza following the ideas of developing self programming computers [12]. Similar to other evolutionary techniques (GA, evolutionary programming, evolutionary strategies etc.), GP uses a computational model of Darwin's natural selection based on the survival of the fittest. According to Darwin, the individuals best adapted to their environments have the highest chances of sharing their genes into the next generations.

Koza used LISP expressions to encode computer programs as individuals which the GP algorithm evolved. The sizes and shapes on the parse trees are independent on the problem tackled, and they are constrained only by the capabilities of the computing system used to run the algorithm. Based on Koza's ideas, many other types of chromosome representations have been proposed in the literature (stack-based GP, machine code GP, etc). In order

to cope with various degrees of solution complexity, most of these alternative representations use variable-length chromosomes (same as Koza's algorithm).

Cândida Ferreira introduced Gene Expression Programming [8], a GP variant that uses chromosomes of a GA-like structure. The new representation uses fixed-length chromosomes, represented by strings of symbols, like in the classical genetic algorithms, that encode parse trees of computer programs or mathematical expressions. Although the length of the chromosome is fixed, the sizes and shapes of the expressions parse trees can vary, same as in GP. The expressions can include different mathematical symbols: functions, variables, and constants, which are used to define computational models of real life processes. The symbols set that the algorithm can use in chromosomes is a parameter of the algorithm.

Each GEP chromosome is composed of one or more genes. The number of genes is constant during the evolution process and is equal in all individuals. Genes are also divided into two sections: head and tail. The head section can contain any symbols from the symbols set: functions, constants and variables. The head size is a parameter of the algorithm and is fixed during the evolution process. The tail section contains only terminal symbols—0-arity symbols, usually constants and variables. Its size is computed based on the head size and the maximum arity of the symbols in the symbols set, such that the tail contains enough symbols to complete the parse tree defined by the head. This way, each GEP gene decodes into a correct parse expression tree, and therefore a correct mathematical function. The expression encoded by a chromosome is obtained by linking the expressions obtained from the genes.

The fitness is measured by mean squared error of the models encoded by chromosomes with respect to the observed data. In experiments, we report the prediction error of the models

$$error = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{x_i})^2},$$

where

- x_i is the value in the original data series at moment i,
- $\hat{x_i}$ is the value at moment i predicted by the model.

We also report the ratio of the prediction error over the standard deviation of the series in order to assess the quality of the predictions.

The genetic operators in GEP are very similar to the operators in the classical bit-string genetic algorithm, due to the linear structure of the chromosome. All operators are constrained to leave the structure of the genes

unchanged: only terminals may go in the tail. Basically, mutation acts by randomly choosing a symbol and replacing it with a new one, while recombination randomly chooses a cut point and constructs offspring by exchanging parts between the parents.

Apart from the GA-like mutation and recombination, a new type of operators is defined in GEP – transposition. The main idea behind transposition is that it duplicates portions of code within a chromosome. For a detailed description of the GEP paradigm and a thorough analysis of its behavior, we refer the reader to [8].

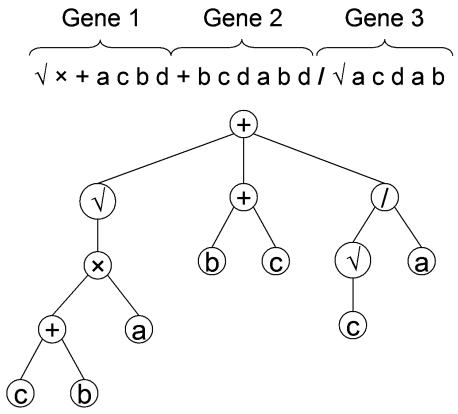

Figure 1: Standard GEP chromosome and the parse tree of the expression it encodes.

As all algorithms in the field of evolutionary computation, GEP has a lot of parameters that need to be set prior to running the algorithm. A crucial parameter that controls the size and implicitly the complexity of the evolved solutions is the number of genes in a chromosome (all chromosomes in a population have the same number of genes). Empirical results suggested that modifying the number of genes had a major impact on the solution obtained by the algorithm [8]. A very large number of genes may lead very well to over-fitting, while a too small number of genes may severly affect the chances to find a well fit model.

Taking this into account and given the complex nature of the problem we tackle, we decided to use the Adaptive Gene Expression Programming algorithm [3]. AdaGEP is a hybrid algorithm that couples a genetic algorithm to GEP in order to automatically identify the appropriate number of genes. Each AdaGEP chromosome is enhanced with a bit string, called “genemap”. Each bit in the genemap corresponds to a gene in the GEP chromosome: if

the bit is set, the respective gene will be decoded and will be a part of the expression decoded from the chromosome, else the gene is ignored (inactive). A gene that is inactive in a chromosome may become active if its bit is flipped in a genetic operation.

Since every GEP chromosome in the population of the algorithm has a genemap attached, we obtain a population of genemaps that regulates the decodification of the chromosomes into mathematical expressions. The genemaps are evolved with a genetic algorithm. Practically, for each GEP generation, a GA iteration is performed. There is a slight difference from the classic GA: the individuals are not evaluated, but they receive as fitness values the fitness of the GEP chromosomes they are attached to. Therefore, the GA iteration involves only performing genetic operations: mutation and crossover.

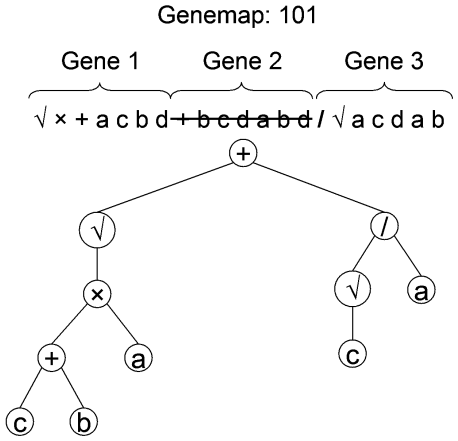

Figure 2: AdaGEP chromosome. The second bit in the genemap is 0, therefore the second gene is inactive and does not participate in the encoded expression.

The two populations coevolve and receive feedback from one another: GEP influences the GA by setting the fitness values, while GA influences GEP by deactivating genes, which further reflects upon the mathematical expressions obtained. In [3] there is a detailed description and analysis of AdaGEP.

4.2.1 Experimental setup

We used our AdaGEP extension [3] implemented for the `gep` package of the framework `ECJ` for the experiments described in this work. The parameters of

ECJ is an open-source evolutionary computation research system developed in Java at George Mason University's Evolutionary Computation Laboratory and available at

the algorithm are described in the following. The maximum number of genes in the GEP chromosomes was set to 6. Therefore, each AdaGEP chromosomes has associated a genemap of 6 bits. The size of the head of a gene was 5 symbols, the population size 1000, as termination criterion for the algorithm we used a maximum number of generations of 200. The operator rates were left at the default values provided by the framework. The function set included the arithmetic operators $\{+, -, *, /\}$ and also trigonometric functions $\{\sin, \cos\}$.

The selection scheme used was fitness proportionate selection, enhanced with elitist survival of the best 10% of the individuals in each generation onto the next. The algorithm is driven by the feedback it receives from the population by means of the fitness values of the individuals; individuals with smaller errors are favored. The genetic algorithm that evolves the genemaps used a mutation rate of 0.001 and a crossover rate of 0.65.

A very important parameter for algorithms for time series analysis is the number of historical values upon which they base the prediction. It is usually referred to as window size, and the values are usually sampled contiguously from past data. Finding the appropriate window size for a model is an optimization problem itself, and few attempts to tackle it are encountered in the literature. Given enough past values, the gene expression programming algorithm should discover by evolution the important variables and include them in the model. Nevertheless, in this study we take on a brute-force approach, relying on information regarding the nature of the series we study. This way, we perform experiments for all window sizes in the interval $[1, 12]$ and report here the best models encountered. Evidently, for each window size, 50 runs are performed.

5 Experimental Results

The time series studied are represented in Figures 3 and 4 and are denoted respectively by S_1 and S_2. They are the series of mean annual and mean monthly precipitation registered in the period 1965 - 2005 at Medgidia meteorological station, situated in Dobrudja region, in the South - East of Romania.

These two series were chosen in order to test the limits of the methods since they are different: S_2 is much longer than S_1 and it presents a seasonal behavior, discussed in the following.

Statistical analysis on the time series is performed applying the tests and procedures described earlier. The results are presented in the following.

- For S_1:

http://cs.gmu.edu/ eclab/projects/ecj/

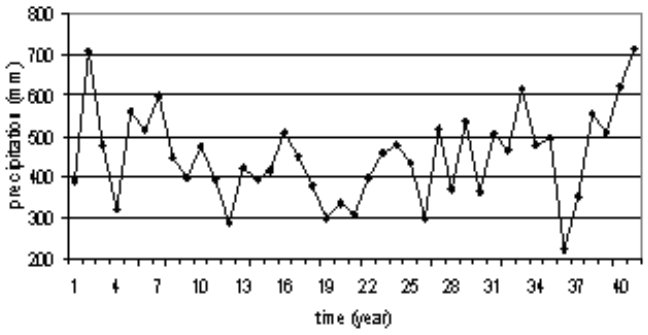

Figure 3: S_1: The mean annual precipitation (January 1965 - December 2005).

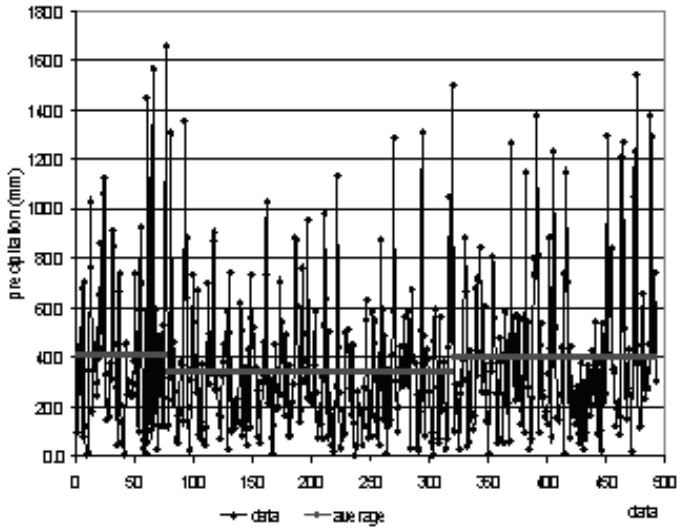

Figure 4: S_2: The mean monthly precipitation (January 1965 - December 2005).

The series is normally distributed, independent and homoscedastic [2]. Therefore, further statistical analysis of this series in unnecessary.

- For S_2:

  First, it must be noted that the precipitation levels recorded in May 1972 and June 1990 are outliers and were eliminated from the series.

A. The series does not have a normal distribution. The Q-Q plot diagram (Fig. 5) shows that the observed values are not distributed along the straight line that represents the theoretical normal distribution.

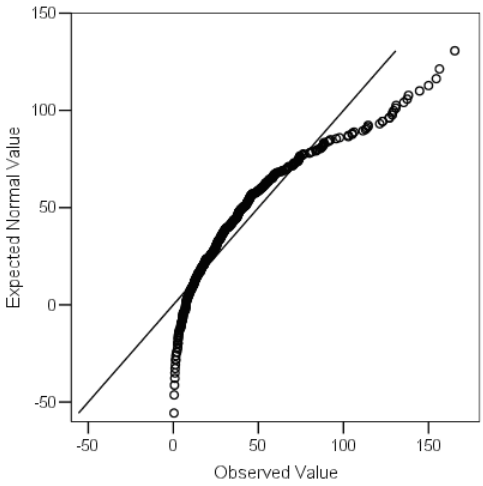

Figure 5: Q - Q plot of S_2.

A new series, denoted by S_2_T, was constructed by applying a Box-Cox transformation $Z_t = \frac{X_t^\lambda - 1}{\lambda}$, with $\lambda = 0.39$.

The Jarque - Bera test applied to this series leads us to accept the hypothesis that the series is normally distributed. Also, the histogram associated to S_2_T confirms the normality (the curve represents the chart of theoretical standard Gaussian distribution) (Fig. 6).

B. S_2 and S_2_T are correlated, since there are values of ACF outside the confidence interval, at the confidence level of 95% (Figs. 7 and 8).

C. Change point analysis gave different results. The application of the Pettitt test to S_2 and S_2_T (Fig. 9) lead us to accept the hypothesis that there is no break in the time series, at a confidence level of 95%.

The same conclusion is obtained by the Buishard test. To confirm this, we include the Bois ellipse associated to this test.

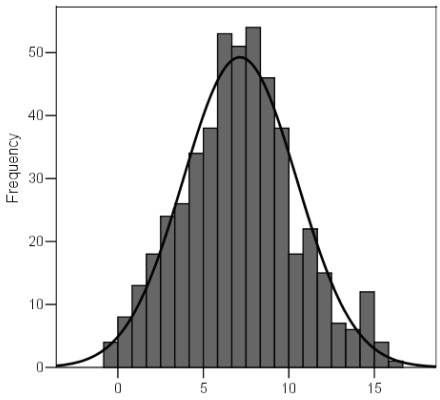

Figure 6: Histogram of S_2_T.

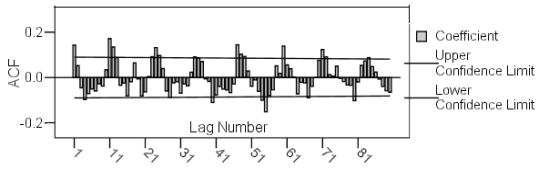

Figure 7: Autocorrelogram of S_2.

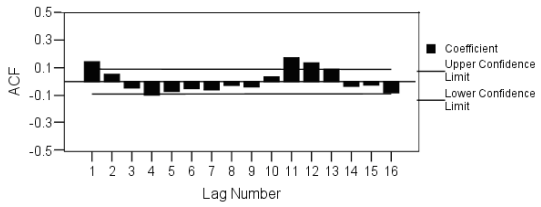

Figure 8: Autocorrelogram of S_2_T.

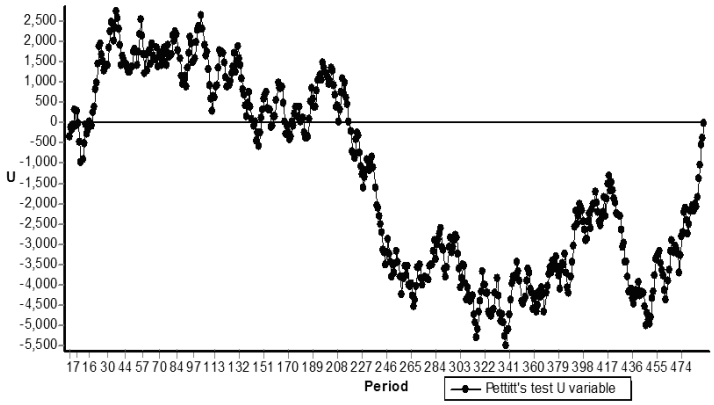

Figure 9: Pettitt test for S_2.

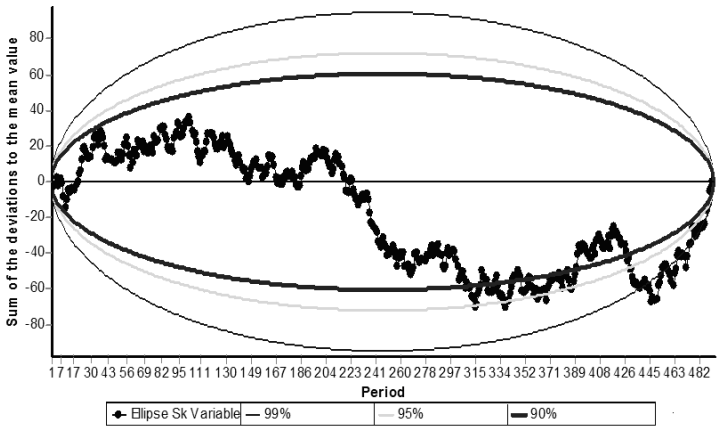

Figure 10: Bois' ellipse.

Since the previous tests are applicable only in the case of a single break in the time series, we further applied Hubert's segmentation procedure to test whether there exist multiple change points in the series. The null hypothesis is rejected for both S_2 and S_2_T. Two break points were determined: in April 1971 and July 1991. This leads to the partition of series S_2 into three subseries, denoted respectively by:

- S_21 - the subseries up to the first break point,
- S_22 - the subseries consisting of the values between the two break points,
- S_23 - the subseries between the second break point and the end of the S_2.

The values of some descriptive statistics of these series are given in Table 1.

Table 1: Descriptive statistics

Series	S_2	S_21	S_22	S_23
min	0.4	0.9	0.4	0.7
max	165.4	156.4	135.4	154.5
mean	37.49	40.8	34.12	41.29
median	29.6	32.4	28.35	30.35
variance	950.77	1153.11	685.38	1045.77
std.dev.	30.83	33.96	26.18	32.34

The CUSUM chart reveals a change point in S_2 (Fig. 11) and also in S_2_T.

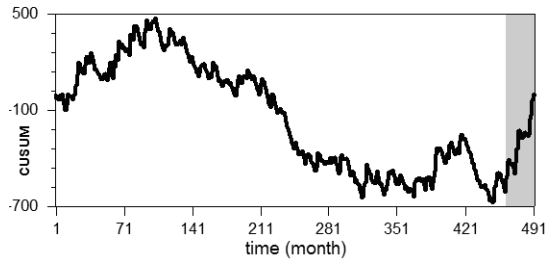

Figure 11: CUSUM of S_2.

D. Since Bartlett test is sensitive to deviances from normality, it was applied to S_2_T. The value of the associate statistic was:

$$T = 1.1166 < 5.99 = \chi^2_{0.05,2}.$$

Therefore the homoscedasticity hypothesis can be accepted at a significance level of 5%.

6 Models

Models obtained with the Box-Jenkins methods are presented next, as well as the symbolic models obtained in AdaGEP experiments.

The models resulted from the evolutionary approach were obtained in experiments of 50 independent runs each, with the same parameters for all runs. All models obtained with GEP presented here were chosen from among the best models (in terms of prediction error) that have independent, normally distributed and homoscedastic residuals.

For the monthly data, the best models were found for the window size of 5, while for the annual data the best model reported was encountered in a run with a window size of 4.

6.1 Models for S_1

We saw that S_1 is Gaussian, independent, uncorrelated and homoscedastic, thus it is a Gaussian noise. Therefore, it is not the case to look for a good model of ARIMA type.

Using AdaGEP, the best solution had the prediction error of 64.17 and the ratio of the prediction error over standard deviation is 0.69.

Combining an AdaGEP model with a backshift operator, results are improved. For example, starting with a model for which the prediction error was 111.874, the final model had the prediction error 45.94 (Fig. 12).

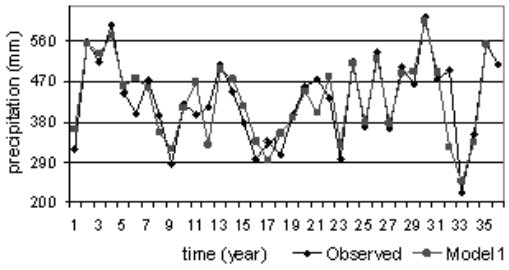

Figure 12: Combined model for S_1.

6.2 Models for S_2

Since S_2_T is Gaussian, the following solutions are proposed:

1. After the mean extraction, a model of ARMA(2,2) type was determined (Fig.11). Its equation is:

$$Z_t = 0.9577Z_{t-1} - 0.9915Z_{t-2} + \epsilon_t - 0.929\epsilon_{t-1} + 0.9914\epsilon_{t-2}$$

where (ϵ_t) is a white noise with the variance 0.9517;

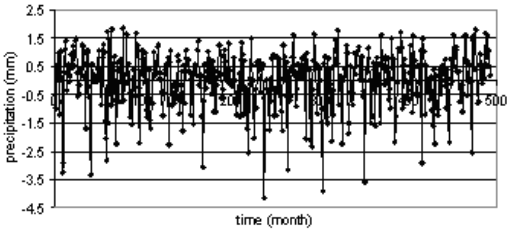

Figure 13: ARMA(2, 2) model for S_2_T.

2. After the elimination of seasonal factors (Table 2) from S_2_T, the new series is Gaussian noise, therefore it is not necessary to determine a better model of ARIMA type.

Table 2: Seasonal factors.

Month	Seasonal Factor (%)
January	79.7
February	82.6
March	83.9
April	97.6
May	111.9
June	130.1
July	126.6
August	106.5
September	96.4
October	90.2
November	97.6
December	96.9

The best solution obtained with AdaGEP for S_2_T was not satisfactory (Fig.14); it did not follow the trend of the series at any point and the amplitude of the model was much smaller than that of the real data.

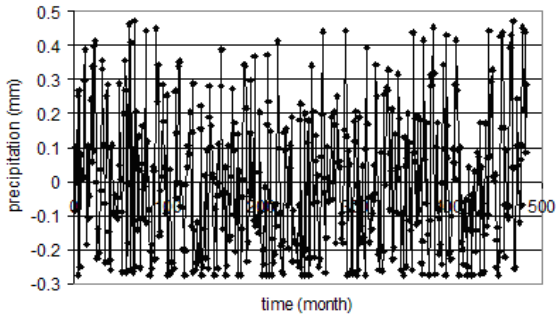

Figure 14: AdaGEP model for S_2_T.

These poor results may be due to the length of the time series (we recall it consists in monthly mean values over 40 years) and the fact that its overall properties describe it as hard to model as a whole, given that it consists of different processes. Also, another possible cause is the large variation of precipitation recorded over some months, which is a characteristic of the entire region.

In order to obtain a better fit, the seasonal factors were eliminated and the resulted series was further modeled with AdaGEP. The model obtained this time for a window size of 12, better fit the data than the previous one (Fig. 15), but it remains worse than the ARMA(2,2) model in terms of prediction error. Its prediction error is 10.05, and the ratio between prediction error and standard deviation is only 0.41.

6.2.1 Models for S_21

Since Hubert segmentation procedure indicated the existence of three different segments for series S_2, we model each subseries independently with classical ARIMA and with the adaptive gene expression programming algorithm.

First, in order to model S_21, the ACF and the Partial ACF were studied. Some of their values lie outside the confidence limits at the level of confidence of 95%.

After a Box-Cox transformation, with $\lambda = 0.39$, the normality of S_21 was reached. The form of ACF associated to the new series, S_21_T, was of damped sine and the values of PACF were inside the confidence level, excepting the

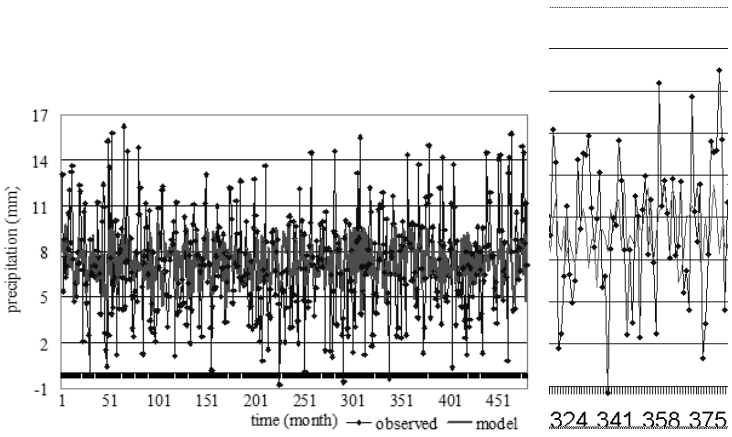

Figure 15: AdaGEP model for S_2_T without seasonality. The picture in the right is a detail of a portion of the time series. It can be noted that the model follows the trend in the original time series, although the amplitude of the values predicted by the model is lower than the real amplitude.

fourth. Thus, the chosen model was of moving average type.

Using Akaike's criterion for the model selection, the best model was:

$$Z_t = \epsilon_t - 0.2242\epsilon_{t-4},\ t \in \overline{5,76},$$

with $(\epsilon_t)_{t\in\overline{5,76}}$ a white noise.

In Fig.16 it can be observed that the model is very good, since the data and the estimated values are practically superimposed in the majority of cases. The charts of the best models obtained for S_21_T using AdaGEP, respectively AdaGEP followed by the application of a backshift operator are presented in Figs. 17 and 18. Their corresponding prediction errors were respectively 27 and 26.238. The ratio between the prediction error and the standard deviation for the AdaGEP solution is 0.86. A slight decrease in the error was accomplished by using the backshift operator.

6.2.2 Models for S_22

A satisfactory model of ARIMA type wasn't found for this subseries. In Fig. 19 we present the chart of the best model obtained using AdaGEP. It has the prediction error of 26.13 and the ratio of prediction error over the standard deviation of 1.01.

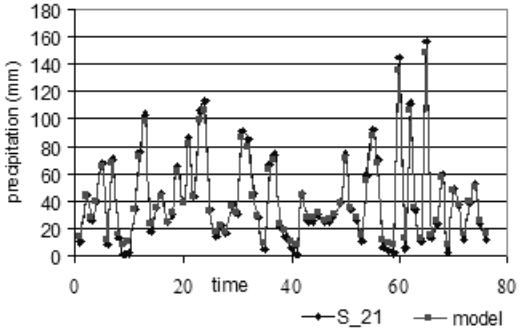

Figure 16: MA(4) model for S_21_T.

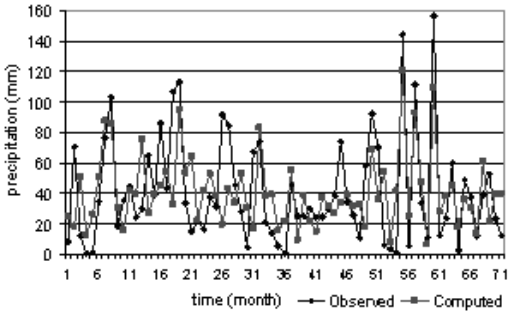

Figure 17: AdaGEP Model for S_21_T.

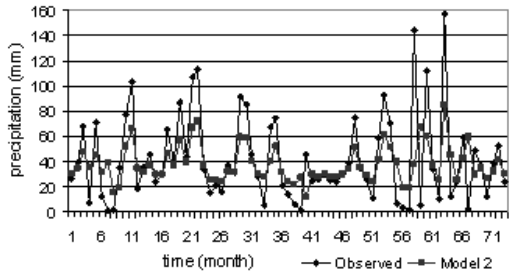

Figure 18: Combined model for S_21_T.

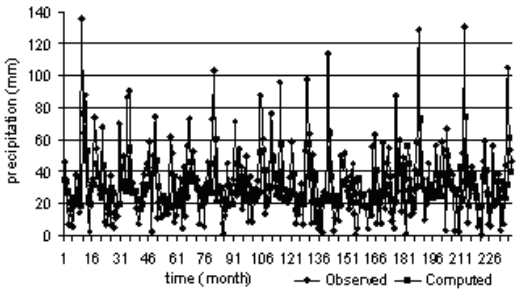

Figure 19: AdaGEP model for S_22.

6.2.3 Models for S_23

The model determined using Box-Jenkins methods was an MA(11):

$$X_t = \epsilon_t - 0.6399\epsilon_{t-11},\ t \geq 12,$$

where $(\epsilon_t)_{t\geq 1}$ is a white noise.

Models obtained with AdaGEP performed similarly, in terms of prediction error, with the moving average model. Still, as expected, an improvement of the calculation error was observed after the application of a backshift operator to the AdaGEP model (Fig. 20).

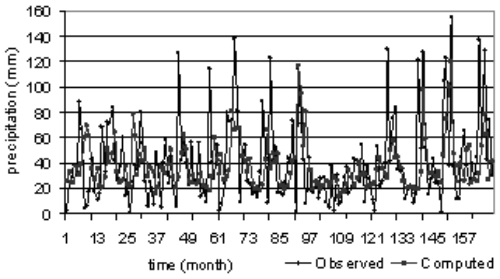

Figure 20: Combined model for S_23.

7 Conclusions

We presented two distinct approaches to time series analysis. The overall methodology included a thorough statistical analysis of the data and classi-

cal and heuristic methods to model the time series. The study shows the suitability of this hybrid approach that combines classical statistical methods to detect changes in the time series with computational intelligence modeling methods. The gene expression programming algorithm is confirmed as a fair competitor for classical ARMA models. While the classical approach adopted in this paper leads to ARMA models that are fairly simple in comparison to the complex expressions evolved by gene expression programming, it lacks the robustness of the evolutionary approach by means of GEP. GEP may be used regardless of the statistical properties of the time series and it may lead to complex non-linear models that fit well the data.

An empirical observation on the heuristic approach is that is seems to work better on smaller time series. This may lead to overfitting, therefore future investigations will be conducted in this direction. Still, a possible explanation for this behavior may be that long precipitation time series cover a long horizon of time, so it is highly possible that some important events appear in the environment of the time series that cause it to shift and change its model. This is in accordance to the methodology we used, where an important step is detecting changes in the time series prior to fitting the models. Also, interesting results were obtained combining the models obtained by the heuristic approach with backshift operators. This is also o direction for future investigations. Future research will deal with evolving teams of predictors using adaptive gene expression programming in order to improve the robustness of the predictions and also extending the adaptive behavior of the algorithm in order to automatically find the best settings for more of its parameters (e.g. population size, operator rates).

References

[1] A. Agapitos, M. Dyson, J. Kovalchuk, S.M. Lucas, *On the Genetic Programming of Time Series Predictors for Supply Chain Management*, in Proc. of the 10th Annual Conference on Genetic and Evolutionary Computation, M. Keijzer, Ed. GECCO '08. ACM, New York, NY, 2008, pp. 1163–1170.

[2] A. Barbulescu, E. Bautu, *Meteorological Time Series Modeling Based on Gene Expression Programming*, Recent Advances in Evolutionary Computing, WSEAS Press, 2009, pp. 17–23.

[3] E. Bautu, A. Bautu, H. Luchian, *AdaGEP - An. Adaptive Gene Expression Programming*, Proceedings of the Ninth international Symposium on Symbolic and Numeric Algorithms For Scientific Computing (September 26–29, 2007), SYNASC, IEEE Computer Society, pp. 403–406.

[4] A. Buishard, *Tests for detecting a shift in the mean of hydrological time series*, Journal of Hydrology, Vol.**73** (1984), pp. 51–69.

[5] R. A. Davis, T. C. M. Lee, G. A. Rodriguez-Yam, *Structural breaks estimation for non-stationary time series signals*, Journal of the American Statistical Association, Vol. **101**, No. **473** (2006), pp. 223–229.

[6] P. Brockwell, R. Davies, *Introduction to time series*, Springer, New York, 2002.

[7] I. De Falco, A. Della Cioppa, E. Tarantino, *A Genetic Programming System for Time Series Prediction and Its Application to El Nio Forecast*, Advances in Soft Computing, Vol. **32** (2005), pp. 151–162 .

[8] C. Ferreira, *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*, Springer - Verlag, 2006.

[9] P. Hubert, *The segmentation procedure as a tool for discrete modeling of hydro-meteorological regimes*, Stochastic Environmental Research and Risk Assessment, Vol.**14** (2000), pp. 297–304.

[10] A. Jann, *Multiple change-point detection with a genetic algorithm*, Soft Computing, Vol. **4 (2)** (2000), pp. 68–75.

[11] M. Basseville, I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Englewood Cliffs, N.J., 1993.

[12] J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press Cambridge, Massachusetts, 1992.

[13] T.L. Lai, *Sequential change point detection in quality control and dynamical systems*, Journal of the Royal Statistical Society, Series B **57** (1995), pp. 613–658.

[14] N. Wagner, Z. Michalewicz, M. Khouja, and R. McGregor, *Time series forecasting for dynamic environments: the DyFor genetic program model*, IEEE Transactions on Evolutionary Computation, Vol. **11(4)** (2007), pp. 433–452.

[15] A. N. Pettitt, *A non-parametric approach to the change-point problem*, Applied Statistics, Vol. **28**, No. **2** (1979), pp. 126–135.

[16] J.R. Sato, P.A.M. Morettin, P.R. Arantes, E. Amaro, *Wavelet based time-varying vector autoregressive modeling*, Computational Statistics & Data Analysis **51** (2007), pp. 5847–5866.

[17] G. W. Snedecor, W. G. Cochran, *Statistical Methods*, 8th Edition, Iowa State University Press, 1989.

[18] D.J. Seskin, *Handbook of parametric and nonparametric statistical procedures*, Chapmann & Hall/CRC, Boca Raton, 2007.

[19] N. Q. Viet Hung, D. T. Anh, *Combining SAX and Piecewise Linear Approximation to Improve Similarity Search on Financial Time Series*, IEEE International Symposium on Information Technology Convergence, ISITC 2007, pp. 58–62.

[20] Y. Zhu, D. Wu, S. Li, *A Piecewise Linear Representation Method of Time Series Based on Feature Points*, Lecture Notes In Computer Science, Vol. **4693** (2007), pp. 1066–1072.

Alina Barbulescu and Elena Bautu
Ovidius University of Constanţa
Faculty of Mathematics and Computer Science
900527 Constanţa, Bd. Mamaia 124
Romania
e-mail: alinadumitriu@yahoo.com, ebautu@univ-ovidius.ro