Time Series Modeling Using an Adaptive Gene Expression Programming Algorithm

Alina Bărbulescu and Elena Băutu

Abstract**—**Meteorological time series are characterized by important spatial and temporal variation. Model determination and the prediction of evolution of such series is of high importance for different practical purposes, even if discovering evolution patterns in such series is a very difficult problem. In this article we describe an adaptive evolutionary technique and we apply it for modeling the precipitation and temperatures collected in a region of Romania. The results are promising for the analysis of such time series.

Keywords— adaptive algorithm, gene expression programming, time series modeling.

I. Introduction

The complexity of the problem of modeling meteorological time series derives from the diversity of phenomena that generally affect the climate (e.g. the greenhouse effect, human influences, solar influences, etc.). Such time series often show non-linear behavior; their analysis constituting a topic of substantial interest in the literature [1], [2], [3], [4]. Meteorological time series are influenced by a multitude of factors, such that their behavior is highly non-linear [5]. Changes in the environment may trigger shifts in the process describing the time series [1], [6]. Classical approaches, such as the linear model, rely on the assumption of a constant data generating process (whose characteristics do not vary with time). Often, they may fail to obtain adequate models due to the nonlinear dynamic behavior of time series, but also due to the lack of adaptation of the methods. This makes the problem very well suited for the use of heuristic methods. They provide means to discover alternative complex, well fit models, without making any assumption on the process under study.

Evolutionary techniques have been used successfully to solve various time series problems. Recurrent neural networks are employed by Badjate and Dudul [2] to obtain short-term, as well as long term predictions of the chaotic sun spots time series. 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 to time series modeling was used in [8], with an emphasis on the accuracy of the predictive solutions discovered by the algorithm. The use of support vector machines for the analysis and short-term prediction of wind speed based on meteorological series was treated in [4].

We focus on a novel adaptive technique to model meteorological time series. The interest in our paper is two-fold. First, we focus on the statistical characteristics of the time series under study. We perform a thorough analysis of the meteorological time series used, which includes the use of statistical techniques to detect whether there exist points in the time series where the process changes. The main interest relies on discovering models for the time series using an evolutionary algorithm that adaptively adjusts some of its parameters during its course.

The evolutionary technique employed in this paper is an improved version of the Gene Expression Programming Algorithm (GEA).

A very important parameter in GEA is the number of genes used by a chromosome, since it affects directly the complexity of the solutions that can be expressed by the individuals in the population. We employ the autoadaptive version of the GEA – AdaGEP, which was initially proposed by us for general symbolic regression problems and we adapt it to be used in the context of modeling time series. Besides performing the standard GEP algorithm, AdaGEP allows the GEA algorithm to find the optimal number of genes used by the chromosomes, through evolution and adaptation, during the run of the algorithm.

The studied time series consist of the mean annual precipitation registered between January 1965 and December 2005, at Medgidia meteorological station, situated in the South – East of Romania, and the time series of the mean annual temperature at Jurilovca station, situated in the Danube Delta. We chose series that concern distinct aspects of the meteorological domain, in order to obtain a better assessment of the applicability of the presented method and to show it performs fine regardless the nature of the data generating processes.

Our article has the following structure:

- considerations regarding the time series modeling problem;
- presentation of basic ideas on the evolutionary technique used to derive the models;

Manuscript received April 29, 2009. This work was supported by UEFISCSU, Romania, under Grant PN II IDEI ID_262/2007 and NatCOMP (PNCDI II 2120/2007).

Alina Barbulescu is with the Ovidius University of Constanta, Faculty of Mathematics and Computers Science, Bd. Mamaia, 124, 900527, Constanta, ROMANIA (e-mail: alinadumitriu@yahoo.com)

Elena Băutu is with the Ovidius University of Constanta, Faculty of Mathematics and Computers Science, Bd. Mamaia, 124, 900527, Constanta, ROMANIA (e-mail: ebautu@univ-ovidius.ro)

- statistical analysis of studied time series;
- experiments and results;
- conclusions and discussions of results and possible directions of future research.

II. Problem formulation

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.

In what follows we shall denote by *n* the selection volume.

The problem that arises is to find a model that fits the time series as well as possible. In order to do it, the first step is to decide how many previous data points are used – the "window size". One must also decide how the past data used by the model is sampled from the original time series.

In this study, we denote the window size by *w*, and we sample the past data at a sampling lag *k* =1. This means that, for example, if *w* = 3, the model will predict the value at a moment t, x_t, using the previous 3 values in the sample, namely x_{t-1}, x_{t-2}, x_{t-3}.

In a more formal manner, we are interested in finding a function *f* that predicts the values of a time series as accurately as possible:

$$\hat{x}_t = f\left(x_{t-1}, x_{t-2}, ..., x_{t-w}\right), t \le n .$$

The accuracy of a model is measured in terms of prediction error:

$$error = \sqrt{\frac{1}{n-1}\sum_{t=1}^{n}\left(x_t - \hat{x}_t\right)^2} .$$

Better models are those with smaller prediction error.

We also can report the ratio of prediction error over standard deviation as a measure of the prediction quality in a model.

Finding a function that fits the data is actually an inverse problem, since there may exist more than one solution to it – making it a well-suitable candidate for a heuristic approach. We chose to tackle the problem with an enhanced GEP algorithm, described in the next section.

III. Gene Expression Programming

One of the most important goals of artificial intelligence in general is to endow computers with the ability to program themselves. John Koza proposed the most successful attempt of the problem of automatic programming in [9], where he described the Genetic Programming (GP) paradigm – a generalization of Holland's Genetic Algorithms (GA) [10].

GP belongs to the large family of evolutionary techniques, along side GA, Evolution Strategies, or Evolutionary Programming [9, 10]. These techniques share mechanisms that come from Darwin's theory of evolution – natural selection based on the survival of the fittest. According to the survival of the fittest principle, the individual best adapted to its environment has the highest chance of survival and reproduction, therefore its traits get to live and perpetuate in next generations.

Since Koza's seminal work [9], many variants of GP have been proposed in the literature. The differences among them are, in most cases, triggered by the different representations used to encode the solutions. In this paper, we use the Gene Expression Programming algorithm (GEA), proposed by Ferreira [11], which we briefly describe in the following.

GEP is a flavor of GP that uses a novel representation that takes advantage of both GP and some features of the classical GA [9, 10], and overcomes in this way limitations of the standard GP and GA. In GP, candidate solutions to the problem at hand encoded by the individuals are computer programs expressed as complex hierarchical structures.

In the context of our problem – time series modeling, a candidate solution is a mathematical formula expressed as a composition of mathematical functions, variables, and constants and therefore is well represented as the parse tree of the mathematical expression. GP individuals are obliged to no constraints with respect to their shape or size, other than the physical limitations of the system. In most cases, individuals are subject to a constraint regarding the maximum allowed depth, or the maximum allowed number of symbols (functions, variables, constants).

On the other hand, GEP individuals are fixed size strings of symbols; nonetheless, they encode non-linear expressions. In GEP, individuals are composed of one or more genes of equal length; the number of genes in the chromosomes is constant in all individuals in the population over all generations. In the standard GEA, it is given as a parameter of the algorithms, as is the gene size.

A gene is a linear string of symbols. By symbols we understand mathematical functions (e.g. arithmetic operators like $+, -, *, /$, trigonometric functions, exponential, logarithmic, etc), constants or variables. The set of symbols at the algorithm's disposal is a parameter of the algorithm. Every gene encodes a mathematical expression expressed as an expression tree.

Ferreira proposed a special syntax for GEP genes that ensures the validity of the de-codification process. A GEP gene is structured in two parts, named "head" and "tail". The tail is constrained to contain only constants or variables, whereas the head may contain any symbol. If we denote the head's size by h and the tail's size by t, the relation:

$$t = h(n-1)+1 ,$$

must hold, where n represents the maximum arity of the functional symbols used by algorithm.

This rule is a guarantee that each GEP gene decodes into a correct expression tree, i.e. a correct mathematical function. Fig. 1 presents a possible GEP individual for the time series modeling problem.

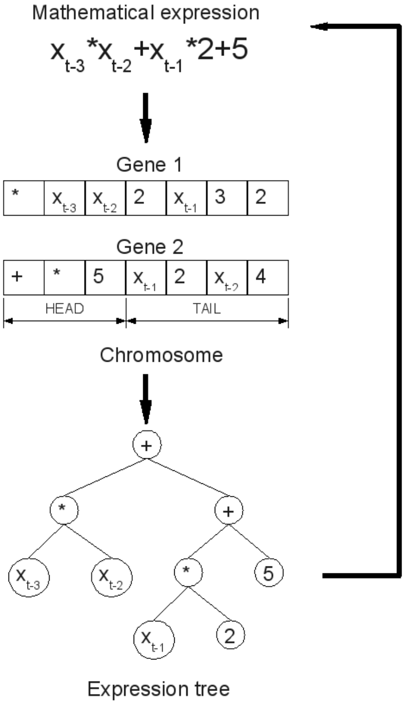

Mathematical expression
x_{t-3}*x_{t-2}+x_{t-1}*2+5
Gene 1
* | x_{t-3} | x_{t-2} | 2 | x_{t-1} | 3 | 2
Gene 2
+ | * | 5 | x_{t-1} | 2 | x_{t-2} | 4
HEAD
TAIL
Chromosome
Expression tree

Fig. 1. GEP individual

In the process of decoding a GEP individual, the expressions encoded by the genes are linked together by means of a linking function. The linking function is also a parameter of the algorithm. It depends very much on the type of problem, but usually it is the addition – as is the case of the work presented here. In the case of Boolean problems, the most used is logical AND.

The linear structure of GEP chromosomes allows the operators in GEP to perform structural changes similar to those performed by classical GA operators. All operators are implemented so as to respect the rule that enforces only terminal symbols in the tail of each gene. The unary operators defined in GEP include mutation and transposition.

The mutation operator changes a randomly chosen symbol in the chromosome.

There exist three transposition operators defined in standard GEP. They work by duplicating sequences of genetic code in a chromosome; the differences among them come from the selection way of the code that is to be duplicated (called *transposon*). The transposon may be any subsequence of the chromosome (IS transposition); it may be selected to begin with a function (RIS transposition), or may consists of an entire gene (gene transposition).

As binary operators, GEP has three kinds of crossover operators – all inspired from their GA counterparts: one-point crossover, two-point crossover and gene crossover. Each of them acts by randomly picking two parents from the population. Then, one point crossover randomly picks a position and swaps the genetic material downstream between the two parents. Gene crossover swaps entire genes among the parents, while two-point crossover swaps the genetic material between two randomly chosen positions. For details on the inner workings of standard GEP operators, see [12].

The criterion used by the algorithm to evaluate the candidate solutions uses the prediction error.

A. Adaptive Gene Expression Programming - AdaGEP

The number of genes in a chromosome is one of the most important parameters in GEP. It has the same value for all the individuals in a population and is constant throughout a run of the algorithm. This is a rather hard constraint, since it controls the shape and the size of the solutions evolved.

Determining the optimum number of genes is an empirical process very much based on the intuition and the experience of the person performing the experiments. Incorrect setting of this parameter may cause the algorithm to fail in finding the true model for the time series: if the resulting length of the chromosome is too small, the complexity of the potential solutions encoded is severely limited. On the other side, if the resulting number of symbols in a chromosome is too big, the search space of candidate solutions increases extremely, such that the best model may not be encountered by the algorithm in proper time.

We use AdaGEP [13], an algorithm that overcomes this issue by identifying the appropriate number of genes automatically. AdaGEP uses an adaptive gene deactivation mechanism inspired by genetic algorithms. Each AdaGEP chromosome is enhanced with a bit string, called "genemap".

The genemap size is equal to the number of genes in a GEP individual. Each bit in the genemap corresponds to a gene of the chromosome and it controls whether that gene is used during chromosome decoding. If a genemap bit is set, the decoding process interprets the corresponding gene as in the classical GEP. If its value is 0, the decoding process ignores the corresponding gene. In this case, we call the gene "deactivated", since it has no effect on the mathematical model encoded by the chromosome.

For example, if the genemap is 011, the first gene is deactivated and the AdaGEP chromosome

```
012345678 012345678 012345678
*+x*xx1xx *-x*x3x2x /x+3x1xx5
```

decodes into $2x^2 + \frac{x}{3+x}$. The classical GEP decoding process would have resulted in the expression $4x^2 + \frac{x}{3+x}$.

In AdaGEP, we obtain a population of genemaps, parallel to that of the GEA. We allow the genemaps to evolve similarly to the population of a classic genetic algorithm. On every GEP iteration, a GA iteration is performed on the genemaps: we apply mutation and crossover on the population of genemaps as they are defined in classical GA.

A genemap survives the selection process only if its corresponding chromosome survives. Thus, AdaGEP allows each chromosome to filter out genes that are not useful during evolution. The hybrid algorithm obtained co-evolves two separate populations: the population of mathematical models encoded by the GEP individuals and the population of genemap encoded by bitstrings, that dictate which genes are to be decoded and which ones are to be ignored in the GEP individuals.

Resuming, the AdaGEP algorithm has the following steps:

1. Create the initial population of individuals (randomly).
2. Evolve individuals with GEP specific operators (crossover, mutation, transpositions).
3. *Apply Gene Map Evolution operator on the population of genemaps.*
4. Evaluate each *AdaGEP individual* over the set of fitness cases.
5. Select the next generation individuals (by roulette wheel selection):
    a. the genemaps are assigned to the fitness value of the individual they are attached to;
    b. *survival of GEP chromosome implies survival of its genemap also!*
6. Go to 2 if the stop criterion is not met.

B. Parameter Settings

The experiments use the AdaGEP extension implemented for the `gep` package of ECJ[1]. We perform runs of the standard GEP algorithm and of the adaptive version presented in order to establish the usefulness of the adaptive approach in the context of time series modeling. In each experiment, 50 independent runs for each setup have been performed.

AdaGEP used a number of genes of 10 per chromosome. This means that the maximum number of genes to be used by the chromosome is 10, but during the run of AdaGEP, the actual number of genes used by each chromosome is adaptively adjusted,. The head size of a gene was set to 5, the population size was set to 200, and the maximum number of generations the algorithm is allowed to run was 1000 (per run).

The operator rates used the default values provided by the `gep` package of ECJ. An important parameter for the algorithm is the function set. Usually, it includes the arithmetic operators. A too large number of functions would lead to an explosion in the search space of possible solutions, which is not desirable [12]. Since the time series under study come form meteorological domain, it is expected they some cyclical behavior. Therefore, the function set used in our experiments consisted of $\{+,-,*,/,\sin\}$, where division is implemented as a Koza style protected operator [9].

Finding the optimum window size is an optimization problem by itself and there exists no precise algorithm to compute it. Since this is not the main purpose of our article, we do not employ a special algorithm to decide on a specific window size. Instead, we take on a brute-force approach: we perform experiments for all window sizes in the interval $w \in \{1,2,3,4,5,6\}$ and the lag $k = 1$ and report the best model over all. Moreover, the nature of the search process employed by GEP allows it to identify automatically the variables that are most useful to estimate future values among the past n input variables.

IV. Data analysis

In order to determine the characteristics of each of the series under observation, the following procedures and statistical tests were used:

1. Kolmogorov – Smirnov, Jarque - Bera tests or Q-Q plot – to test the normality hypothesis [14];
2. Rank correlation test [14] – to verify the hypothesis whether the series is random;
3. The autocorrelation function (denoted by ACF) [15] – to test the hypothesis that the series is uncorrelated;
4. Bartlett or Levene test [16] - to test the homoscedasticity hypothesis;
5. Buishard [17] and Pettitt [18] tests, and Hubert's segmentation procedure [19] – to determine the existence of a break in the time series (break – a point where the model changes.

    CUSUM procedure [20] was also used, to determine the changes in mean in the series.

A. The analysis of Series 1

The series of mean annual precipitation collected at Medgidia station is represented in Fig. 2. We shall refer to it as Series 1. The average precipitation is 449.92 mm and the standard deviation is 109.24.

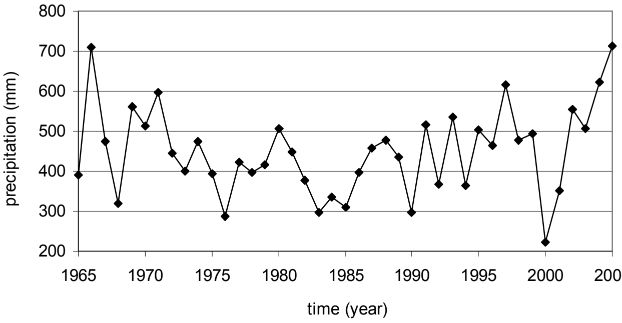


Fig. 2. Series 1

The results of the statistical analysis by means of the previously mentioned test are discussed in the following.

1. The data are normally distributed, since:

- in the Q-Q plot diagram (Fig. 3) the observed values are distributed along the straight line that represents the theoretical normal distribution;

- the p-value associated with Kolmogorov – Smirnov test is bigger than 0.05.

[1] ECJ is an open-source evolutionary computation research system developed in Java at George Mason University's Evolutionary Computation Laboratory and available at http://cs.gmu.edu/˜eclab/projects/ecj/

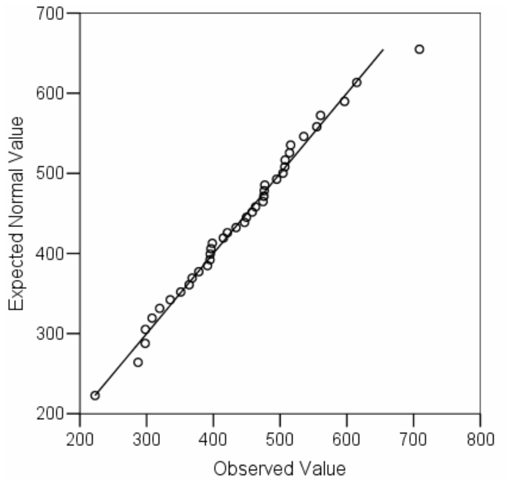


Fig. 3. Q-Q plot of Series 1

2. The series is random.

3. The series is not correlated, since the values of the autcorrelation function are inside the confidence limits at a confidence level of 95% (Fig. 4).

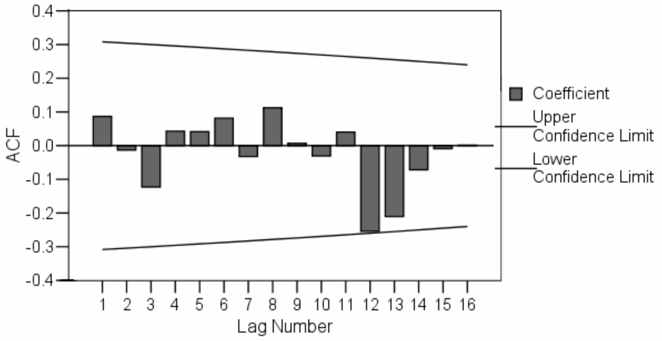


Fig.4. ACF of Series 1

The values of the partial autocorrelation functions (Partial ACF) are also inside the confidence limits at a confidence level of 95% (Fig. 5).

4. Dividing the data in two parts (the first 20 and the last 21 values), and calculating the value of the statistic X^2, in Bartlett test, we obtain:

$$X^2 = 0.7298 < \chi^2_{0.95}(2-1) = 3.84,$$

where $\chi^2_{0.95}(2-1)$ is the quintile value of χ^2 function, with 1 degree of freedom, at a significance level 95%.

Thus, the hypothesis that the time series is homoscedastic is accepted.

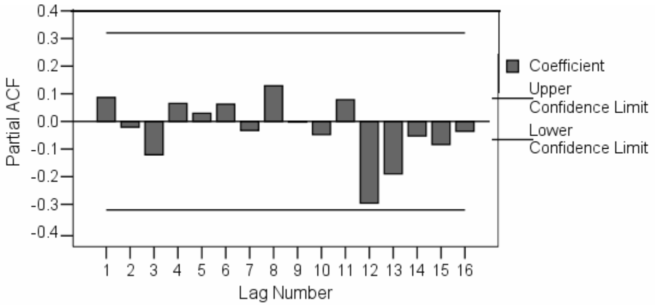


Fig.5. Partial ACF of Series 1

5. After the application of Buishard and Pettitt tests, the hypothesis that there is no break in the series is accepted at the confidence level of 95%. Hubert's segmentation procedure detects a break in 2003.

Also, the CUSUM procedure gives a change point in 2003 (Fig. 6), but since we have only two data after this year, we can not confirm the last hypothesis. As consequence, we eliminate the last two values and we search for the model for the period 1965 – 2003.

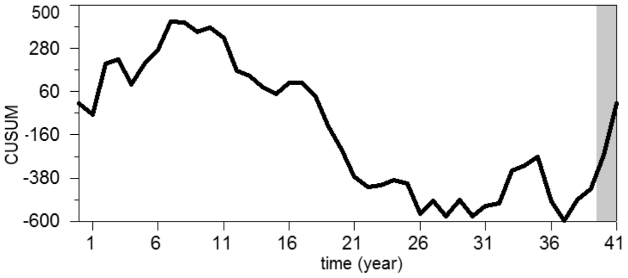


Fig. 6. CUSUM diagram

B. *The analysis of Series* 2

The series of mean annual temperatures (1965 - 2005) at Jurilovca station is represented in Fig.7. We shall refer to it as Series 2. The average temperature is 11^{0}C and the standard deviation 0.66.

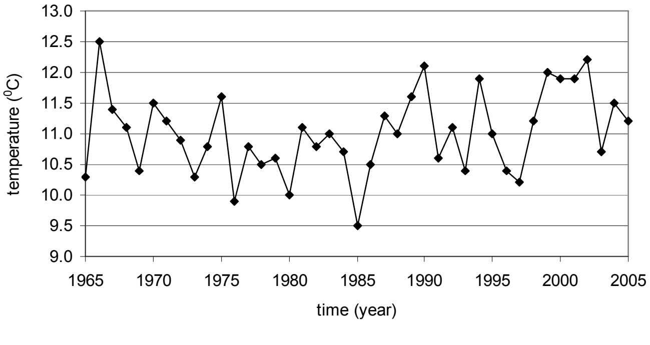


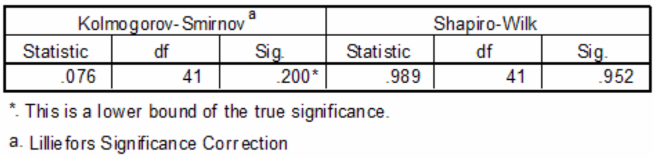
Fig. 7. Series 2

The results of the statistical analysis of Series 2 are presented next.

1. In Table I the results of Kolmogorov - Smirnov and Shapiro – Wilk tests are given, where Statistic represents the values of the corresponding statistic, df represents the degrees

of freedom and Sig. is the signification level of this test. If Sig < 0.05, there is a deviation from normality. As can be noted, there is no deviation from normality.

Table I. Results of Kolmogorov - Smirnov and Shapiro – Wilk tests for Series 2



Kolmogorov-Smirnov[a]			Shapiro-Wilk		
Statistic	df	Sig.	Statistic	df	Sig.
.076	41	.200*	.989	41	.952

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

The histogram associated to Series 2 can be also analysed. (Fig. 8) Taking account on the tests' results we can accept the hypothesis that Series 2 is normally distributed.

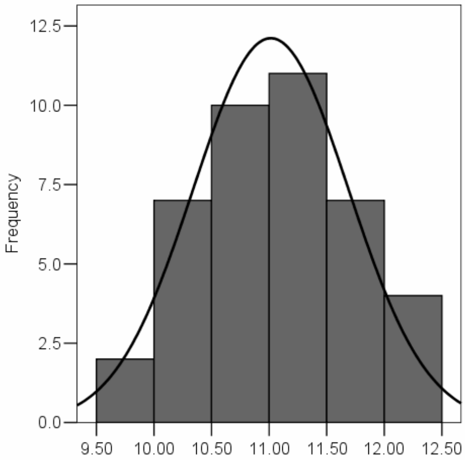
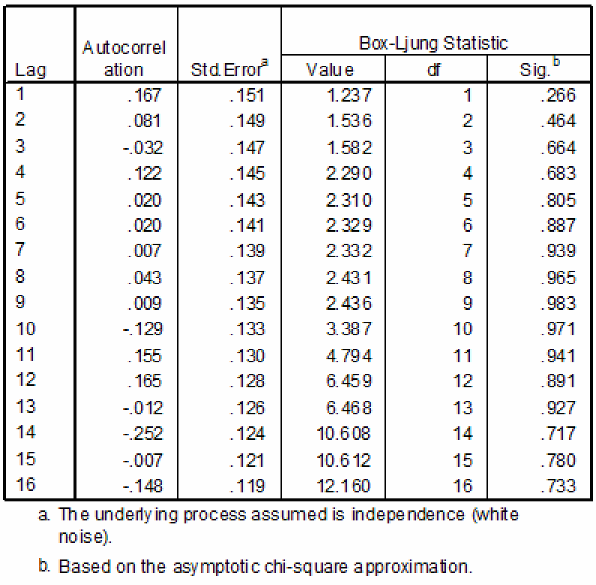


Fig. 8. Histogram of Series 2

Table II. Results of autocorrelation test for Series 2



Lag	Autocorrelation	Std.Error[a]	Box-Ljung Statistic		
			Value	df	Sig.[b]
1	.167	.151	1.237	1	.266
2	.081	.149	1.536	2	.464
3	-.032	.147	1.582	3	.664
4	.122	.145	2.290	4	.683
5	.020	.143	2.310	5	.805
6	.020	.141	2.329	6	.887
7	.007	.139	2.332	7	.939
8	.043	.137	2.431	8	.965
9	.009	.135	2.436	9	.983
10	-.129	.133	3.387	10	.971
11	.155	.130	4.794	11	.941
12	.165	.128	6.459	12	.891
13	-.012	.126	6.468	13	.927
14	-.252	.124	10.608	14	.717
15	-.007	.121	10.612	15	.780
16	-.148	.119	12.160	16	.733

a. The underlying process assumed is independence (white noise).

b. Based on the asymptotic chi-square approximation.

2. The series is random.

3. The results of autocorrelation test are presented in Table II, where the columns contains, respectively: the lag between each two values considered in the calculation of ACF values; the values of ACF; the standard errors when the underlying process is independent; the values of Box – Ljung statistic, the degrees of freedom for which the statistic values were calculated and the significance level.

Since values of Box-Ljung statistic are less than $\chi^2(15)$, we accept the hypothesis that the series is not correlated.

4. The Hubert segmentation procedure and CUSUM detect a break in 1997.

5. Dividing the data sample in two sub - samples, corresponding to the periods before and after the break and applying Levene test, the values of corresponding statistic is:

$$0.306713 < \chi^2_{0.95}(2-1) = 3.84,$$

so the homoscedasticity hypothesis can be accepted.

V. GEP Derived Models

A. Models for Series 1

In this section we present the best models obtained using standard GEP and the adaptive version, AdaGEP.

The overall quality of the solutions was best in the runs that used the window size of 5. Therefore, we resume at the presentation of only the best solutions depicted over all runs by GEP, respectively AdaGEP.

For both series, the standard GEP algorithm performed best in the runs where the number of genes was 5. Nevertheless, the effort to find this out was considerable, since we performed 50 independent runs for each number of genes between 1 and 10.

The first model, obtained using GEP for the mean annual precipitation evolution is presented in Fig.9.

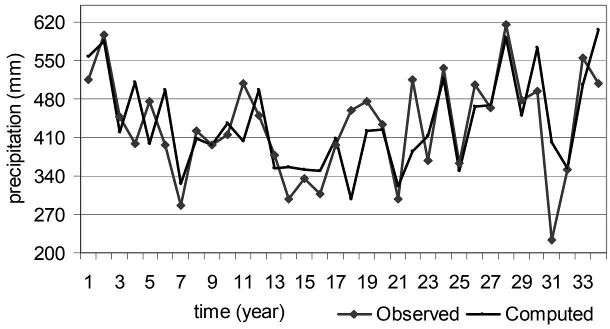


Fig. 9. The first model (GEP)

The number of symbols in this original GEP solution is 43, while the average number of symbols the GEP solutions over the 50 runs is 34.

The residual is normally distributed (Fig.10) and uncorrelated before the lag 12 (Fig. 11), since the probabilities to reject the correlation hypothesis are bigger than 0.8. They are also homoscedastic.

The prediction error was 68.26, and the ratio between the prediction error and the standard deviation was 0.74.

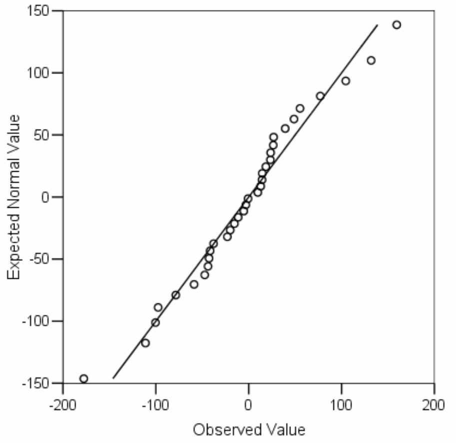


Fig. 10. Residuals' Q - Q plot

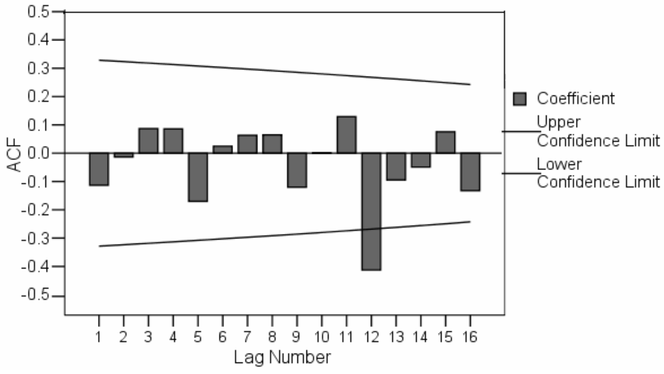


Fig. 11. Residuals' ACF

The second model we present is depicted as best solution encountered by the AdaGEP algorithm (Fig. 12) (over all experiments, each experiment consisting of 50 runs). The algorithm evolved towards this solution that uses only 4 out of the 10 genes in the chromosome.

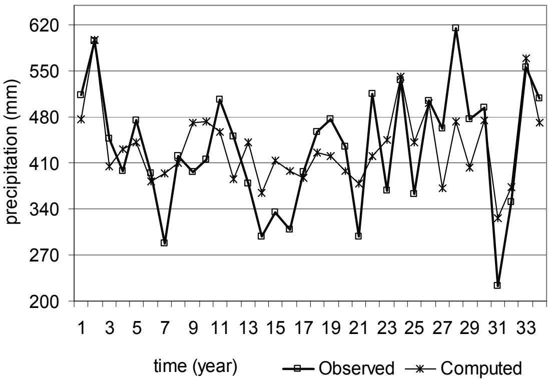


Fig. 12. The second model (AdaGEP)

It may be noted that AdaGEP chromosomes are similar to the real DNA code of living organisms, which contains large sequences of unused code. The number of symbols in this solution was 35, while the average number of symbols over all 50 AdaGEP solutions was 20.

The residual has the same properties as in the previous case.

The prediction error was 64.17, and the ratio between the prediction error and the standard deviation was 0.69.

It is interesting to note that while obtaining a better fit, the AdaGEP algorithm identified the optimal number of genes to be used by GEP individuals, which is close to the value (of 5) "manually" found in the repeated standard GEP experiments in the first phase.

It is interesting to note that on average, AdaGEP solutions used fewer symbols, leading to fewer function evaluations, and therefore a reduction in the algorithm's running times. Over all runs for Series 1, the mean number of genes in the best-of-run solutions was 3.5, the mean number of symbols 20 and the standard deviation of the number of symbols 6. The number of symbols of GEP best-of-run solutions had a mean of 34, with a standard deviation of 7.

B. *Models for Series* 2

Since the Series 2 presents a break in 1987, three different models were searched: for the entire Series 2 and for the subseries before 1987 (denoted by Series 2_1) and after 1987 (denoted by Series 2_2).

As in the previous set of experiments, for each window size (up to 6), and for each number of genes, 50 independent runs are performed of the standard GEP. Also, for each window size, 50 independent runs of AdaGEP with a maximum number of genes of 10 are performed.

We present only the best solutions encountered in the AdaGEP run, since it is slightly better than that encountered by standad GEP in all the experiments that used it. The interesting fact is that the adaptive hybrid algorithm reaches the conclusion that the suitable number of genes to be used by a chromosome, discovered by AdaGEP, is 5. It is consistent with what we found when running standard GEP for all possible number of genes up to 10.

The best solutions over 50 independent runs of AdaGEP were obtained respectively for a window size of 4 for Series 2 and 5 – for Series 2_1 and 2 – for Series 2_2. They are presented in Figs. 13 - 15.

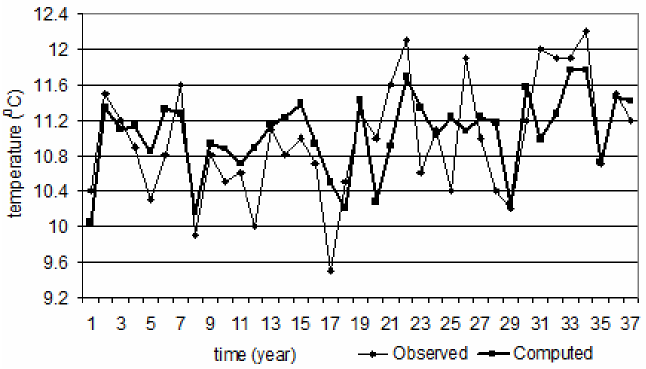


Fig.13. Best AdaGEP Model of Series 2

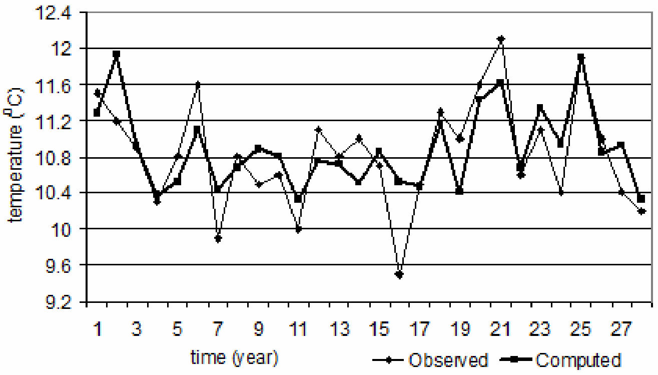


Fig.14. Best AdaGEP Model of Series 2_1

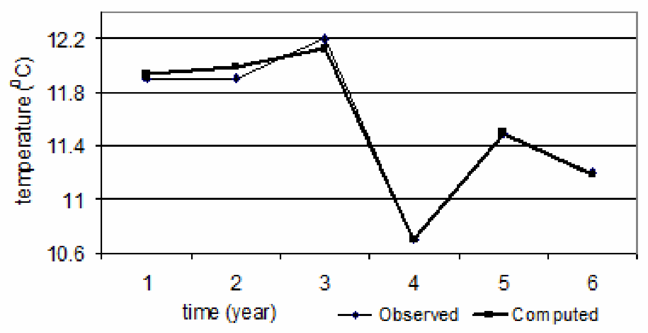


Fig.15. Model of Series 2_2

The mean squared errors were respectively: 0.2391, 0.1501 and 0.00252.

In all the cases the residual are normally distributed, independent and homoscedastic. As it was expected for a good algorithm, the models obtained for the subseries are better than the model for the entire series.

VI. Conclusions

This study confirms the suitability of GEP and AdaGEP to the time series modeling problem. Although the improvements in the quality of the solution are not so impressive, the advantage of using the adaptive version of GEP over the classical version resides in the potential shown for obtaining solutions in a less complex shape and also a significant reduction in the number of fitness evaluations, and consequently in the running time. The coevolving gene map population acts upon the GEP population it implicitly imposing parsimony upon the evolved solutions.

In a previous study [21], GEP was used also to model the long series of mean monthly precipitations January 1965 – December 2005 and the results were comparable with those obtained by GEP. An empirical feature of the GEP approach is that it seems to work better on shorter time series. A possible reason for this behaviour may be the dynamic characteristics around data that concerns weather in general, which coincides with our intuition that there exist points in meteo-hydrological time series when the underlying process changes. Short time series are less likely to contain such change points. Further investigations will follow this direction.

Further research includes evolving teams of individuals to be used as an ensemble model of time series, in order to improve the robustness of the models. Also, AdaGEP will be extended to adaptively find appropriate operator rates for GEP and for the embedded GA.

References

[1] A. Busuioc, H. von Storch, “Conditional stochastic model for generating daily precipitation time series”, *Climate Research*, Vol. 24, 2003, pp. 181–195

[2] S. L. Badjate, S. V. Dudul, “Multi Step Ahead Prediction of North and South Hemisphere Sun Spots Chaotic Time Series using Focused Time Lagged Recurrent Neural Network Model”, *WSEAS Transactions On Information Science And Applications*, Issue 4, Vol. 6, 2009, pp. 684-693

[3] F. Karim, A. Izani, M. Ismail, M. Ashaque Meah, “Numerical Simulation of Indonesian Tsunami 2004 at Penang Island in Peninsular Malaysia Using a Nested Grid Model”, *International Journal of Mathematical Models and Methods in Applied Sciences*, Issue 1, Vol. 3, 2009, pp. 1-8

[4] K. Sreelakshmi and P.R. Kumar, “Short term wind speed prediction using support vector machine model”, *WSEAS Transactions. on Computers,* Issue 11, Vol. 7, 2008, pp. 1828-1837.

[5] S. P. Charles, B. C. Bates, I. N. Smith, James P. Hughes, “Statistical downscaling of of daily precipitation from observed and modeled atmospheric fields”, *Hydrological Processes*, Vol. 18 (8), pp. 1373-1394, 2004

[6] D. S. Wilks, R. L. Wilby, “The weather generation game: a review of stochastic weather models”, *Progress in Physical Geography*, Vol. 23, 1999, pp. 329-357

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

[8] A. Agapitos, M. Dyson, J. Kovalchuk, S.M. Lucas, “On the Genetic Programming of Time-Series Predictors for Supply Chain Management”, in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation* (Atlanta, GA, USA, July 12 - 16, 2008), M. Keijzer, Ed. GECCO '08. ACM, New York, NY, 2008, pp. 1163-1170

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

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

[11] C. Ferreira, “Gene Expression Programming: A New Adaptive Algorithm for Solving Problems”, *Complex Systems*, Vol. 13, No.2, 2001, pp. 87-129

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

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

[14] D.J. Seskin, *Handbook of parametric and nonparametric statistical procedures*, Chapmann & Hall/CRC, Boca Raton, 2007, pp.219-220

[15] P. Brockwell, R. Davies, *Introduction to time series*, Springer, New York, 2002, pp. 16 -19, 46, 59

[16] A. Bărbulescu, *Time series with applications*, Junimea, Iasi, 2002 (in Romanian), pp. 85 – 97

[17] T. A. Buishard, “Tests for detecting a shift in the mean of hydrological time series”, *Journal of Hydrology*, Vol.73, 1984, pp. 51-69

[18] A. N. Pettitt, “A non-parametric approach to the change-point problem”, *Applied Statistics*, Vol. 28, No. 2, 1979, pp. 126 - 135.

[19] P. Hubert, J. P. Carbonnel, ”Segmentation des séries annuelles de débits de grands fleuves Africains”, *Bulletin de liaison du CIEH*, Vol. 92, 1993, pp. 3-10

[20] W. Taylor, 2000, Change – Point Analyse Entreprises, Libertyville, Illinois. Available: http://www.variation.com/cpa

[21] A. Bărbulescu, E. Băutu, "ARIMA and GEP models for climate variation", *International Journal of Mathematics and Computation*, Volume 3, Issue J09, 2009 (in print)