Evolving Time Series Forecasting ARMA Models

Paulo Cortez (pcortez@dsi.uminho.pt)
Departamento de Sistemas de Informação
Campus de Azurém, Universidade do Minho,
4800-058 Guimarães, PORTUGAL
Voice: +351 253 510313

Miguel Rocha (mrocha@di.uminho.pt) and José Neves
(jneves@di.uminho.pt)
Departamento de Informática, Campus de Gualtar, Universidade do Minho,
4710-057 Braga, PORTUGAL

Abstract. *Time Series Forecasting (TSF)* allows the modeling of complex systems as "black-boxes", being a focus of attention in several research arenas such as *Operational Research*, *Statistics* or *Computer Science*. Alternative *TSF* approaches emerged from the Artificial Intelligence arena, where optimization algorithms inspired on natural selection processes, such as *Evolutionary Algorithms (EAs)*, are popular. The present work reports on a two-level architecture, where a (meta-level) binary *EA* will search for the best *ARMA* model, being the parameters optimized by a (low-level) *EA*, which encodes real values. The handicap of this approach is compared with conventional forecasting methods, being competitive.

Keywords: ARMA models, evolutionary algorithms, bayesian information criterion, model selection, time series analysis

Introduction

Time Series Forecasting (TSF), the forecast of a chronologically ordered variable, corporals an important tool to model complex systems, where the goal is to predict the system's behavior and not how it works. Indeed, contributions from the arenas of *Operational Research* led to quantitative *TSF* methods that replaced the old fashioned ones, which were primarily based on intuition. More recently, in the last two decades, alternative nonlinear *TSF* approaches loomed, being *Artificial Neural Networks* the most popular ones. Yet, conventional *TSF* methods are still popular, existing several applications where linear estimations are sufficient (Makridakis et al., 1998). However, these models were developed decades ago, where higher computational restrictions prevailed, being the parameters optimized using numerical methods (e.g., *least squares*) which may be trapped in local minima.

On the other hand, *Evolutionary Algorithms (EAs)* are innate candidates for parameter estimation, since they implement a global multipoint search, quickly locating areas of high quality. The use of *EAs* in *TSF* is expected to increase in importance, motivated by advantages such as explicit model representation and adaptive evolutionary search, which prevents them to fall on undesired local minima. In the past, *EAs* have been used with binary encodings for parameter optimization of traditional *TSF* methods such as *Holt-Winters* (Agapie and Agapie, 1997) or *ARMA* models (Huang and Yang, 1995). However, an increasing focus has been set over the use of *real value* genes (Michalewicz, 1996; Rolf et al., 1997; Cortez et al., 2001), since this direct representation is more scalable and allows the definition of richer genetic operators.

Following this trend, a two-level architecture is presented, where a low-level *EA* with real encodings will be used to estimate the *ARMA* coefficients. Furthermore, the model selection stage will be automatized, by means of a meta-level binary *EA*, which will search through the space of all possible *ARMA* models. The whole evolutionary process will be guided by the *Bayesian Information Criterion (BIC)*, a simple *Information Theory* statistic, that prevents overfitting by adding a model complexity penalty.

The paper is organized as follows: first, the basic concepts for *TS* analysis are defined; then, a description of experiments performed on *EAs* is given; next, the meta-evolutionary approach is presented and explored; finally, the results are discussed and compared with conventional *TSF* methods (e.g., *Holt-Winters* or *Box-Jenkins* methodology).

1. Time Series Analysis

A *Time Series (TS)* is a collection of time ordered observations x_t, each one being recorded at a specific time t (period), appearing in a wide set of domains such as *Finance*, *Production* or *Control.* A *TS* model ($\widehat{x}_t$) assumes that past patterns will re-occur in the future. The overall performance of a model is evaluated by a forecasting accuracy measure, namely the *Sum of Squared Errors (SSE)*, *Root Mean Squared Error (RMSE)* and the *Theil's U* statistic, given in the form:

$$\begin{array}{l} e_t = x_t - \widehat{x}_t \\ SSE = \sum_{i=t+1}^{t+L} e_i^2 \\ RMSE = \sqrt{\frac{SSE}{L}} \\ Theil's\ U = \frac{RMSE}{\sqrt{\frac{\sum_{i=t+1}^{t+L} (x_t - x_{t-1})^2}{L}}} \end{array} \tag{1}$$

where e_t denotes the forecasting error and L the number of forecasts.

A common statistical instrument for *TS* analysis is the *autocorrelation* coefficient (within [-1.0;1.0]), which gives a measure of the statistical correlation between a series and itself, lagged of k periods, being computed as (Box and Jenkins, 1976):

$$r_k = \frac{\sum_{t=1}^{s-k}(x_t - \overline{x})(x_{t+k} - \overline{x})}{\sum_{t=1}^{s}(x_t - \overline{x})} \tag{2}$$

where s denotes the *TS* size. Autocorrelations can be useful for decomposition of the *TS* main features (e.g., *trend* and *seasonal* effects) (Figure 1). A *trend* stands for a constant grow (or decline) in the data, being due to factors like *inflation* or *technological improvements.* The *seasonal* factor is found in series with a periodic behavior and it is very common in monthly series (e.g., umbrella sales).

*** insert Figure 1 around here ***

A quite successful *TSF* method is *Holt-Winters (HW)*, which is based on some underlying patterns (e.g., *trended* and *seasonable*) that are distinguished from random noise by averaging the historical values. Its popularity is due to advantages such as the simplicity of use, the reduced computational demand and the accuracy of the forecasts, specially with seasonal *TSs.* The general model is defined by the basic equations (Makridakis et al., 1998):

$$\begin{array}{l} F_t = \alpha \frac{x_t}{S_{t-K}} + (1-\alpha)(F_{t-1} + T_{t-1}) \\ T_t = \beta(F_t - F_{t-1}) + (1-\beta)T_{t-1} \\ S_t = \gamma \frac{x_t}{F_t} + (1-\gamma)S_{t-K} \\ \widehat{x}_t = (Fx_{t-1} + T_{t-1})S_{t-K} \end{array} \tag{3}$$

where F_t, T_t and S_t stand for the smoothing, trend and seasonal estimates, K for the seasonal period, and α, β and γ for the model parameters.

The Box-Jenkins (1976) methodology is another important *TSF* approach, going over model identification, parameter estimation, and model validation. The main advantage of this method relies on the accuracy over a wider domain of *TSs*, despite being more complex, in terms of usability and computational effort, than *Holt-Winters*. The global model is based on a linear combination of past values (*AR* components) and errors (*MA* components), being named *AutoRegressive Integrated Moving-Average (ARIMA)* (the seasonal version is called *SARIMA*). Both *ARIMA* and *SARIMA* models can be postulated as an $ARMA(P, Q)$ one, given in the form:

$$\widehat{x}_t = \mu + \sum_{i=1}^{P} A_i x_{t-i} + \sum_{j=1}^{Q} M_j e_{t-j} \tag{4}$$

where P and Q denote the AR and MA orders, A_i and M_j the AR and MA coefficients, being μ a constant value. The constant and the coefficients of the model are estimated using statistical approaches (e.g., least squares methods). Trended *TSs* require a differencing of the original values. The methodology also contemplates the possibility of some kind of transformation in the original data (e.g., logarithmic variation).

*** insert Table 1 around here ***

*** insert Figure 2 around here ***

To the experiments carried out in this work, a set of eight series were selected (Table 1 and Figure 2), ranging from financial markets to natural processes (Box and Jenkins, 1976; Makridakis et al., 1998; Hyndman, 2003). All series were classified into four main categories, that encompass the majority of the *TS* types, namely *Seasonal and Trended*, *Seasonal*, *Trended* and *Nonlinear*. Each *TS* will be divided into a *training set*, containing the first 90% values and a *test set*, with the last 10%. Only the training set is used for model selection and parameter optimization, being the test set used to compare the proposed approach with other methods.

2. Evolutionary Forecasting Models

The term *Evolutionary Algorithm (EA)* (Bäck, 1996) names a family of procedures (e.g.,*Genetic Algorithms* or *Evolutionary Strategies*), where an evolving *population* composed of a set of individuals, seeks for a good solution to a given problem. Each individual encodes a solution in a string (*chromosome*) of symbols (*genes*), to each a numerical value (*fitness*) is assigned, that stands for the solution's quality.

New solutions are created through the application of genetic operators (typically *crossover* and *mutation*) and the whole process is driven by a stochastic process, inspired in *natural selection*, which favors individuals with higher fitnesses. The *EA* used in this work is given by following pseudo-code:

```
BEGIN
   Population initialization and evaluation
   WHILE (termination criteria is not met)
      Select ancestors to reproduction
      Create new individuals using genetic operators
      Evaluate the new individuals (offspring)
      Select the survivors and add to the next generation
END
```

Two approaches to forecasting, both based on *EAs* with real valued genes, were followed. In the former one, the forecasting model is a linear combination of previous values. Under this scenario, the genes in the chromosome code for the weights by which previous values are multiplied. With the latter, both previous values and errors are taken into account, following a strategy inspired on the *ARMA* models, where the genes code for the coefficients. Both models make use of a *sliding time window* that defines the set of time lags used to build a forecast, also defining the number of the model inputs. A time window will be denoted by the sequence $< k_1, k_2, ..., k_n >$, for a model with n inputs and k_i time lags.

The two models considered are given, in terms of a predefined time window, by:

$$\begin{array}{rl} AR : & \widehat{x}_t = g_0 + \sum_{i \in \{1,...,n\}} g_i x_{t-k_i} \\ ARMA : & \widehat{x}_t = g_0 + \sum_{i \in \{1,...,n\}} (g_i x_{t-k_i} + g_{i+n} e_{t-k_i}) \end{array} \tag{5}$$

where g_i stands for the i-th gene of the individuals' chromosome.

In this work, two genetic operators were adopted:

Arithmetical Crossover - each gene in the offspring will be a linear combination of the values in the ancestors' chromosomes, in the same positions (Michalewicz, 1996). If a_i and b_i are the offspring's genes, and z_i and w_i the ancestors' ones, at the position i, then $a_i = \lambda \cdot z_i + (1 - \lambda) \cdot w_i$ and $b_i = \lambda \cdot w_i + (1 - \lambda) \cdot z_i$, where λ is a random number in the range $[0; 1]$.

Gaussian Perturbation - a *mutation* operator that adds, to a given gene, a value taken from a gaussian distribution, with zero mean; i.e., small perturbations will be preferred over larger ones (Fogel, 1999).

In terms of the *EA's* setup, the initial populations' genes were randomly assigned values within the range $[-1.0, 1.0]$. The population size was set to 50. The fitness of each chromosome was measured by decoding the individual into the forecasting model and measuring the error over all the training patterns ($RMSE_t$).

The selection procedure is done by converting the fitness value into its ranking in the population and then applying a roulette wheel scheme. In this work, the following evolutionary engine was applied: in each generation, 40% of the individuals are kept from the previous generation, being 60% generated by the application of the genetic operators; the *crossover* operator is responsible for breeding $\frac{2}{3}$ of the offspring and the *mutation* one is accountable for the remaining ones; finally, the *EA* is stopped after 1000 generations.

3. Heuristic Approach to Model Selection

The *EA* presented above can be used for parameter estimation. However, the issue of *model selection*, i.e. choosing the best model for a given *TS*, remains. This is strongly related with the choice of the adequate time window (e.g., *SARIMA* models often use the <1, 12, 13> lags for monthly seasonal trended series).

A good model should be able to learn from training data while generalizing to new observations, avoiding *overfitting*. The usual statistical approach is to consider different candidate models, which are evaluated according to a generalization estimate. Several complex estimators have been developed (e.g., *K-fold validation* or *Bootstrapping*), which are computationally burdensome (Sarle, 1995). A reasonable alternative is the use of simple statistics that add a penalty, that is a function of model complexity, such as the *Bayesian Information Criterion (BIC)* (Schwarz, 1978):

$$BIC = N \cdot ln(\tfrac{SSE}{N}) + p \cdot ln(N) \qquad (6)$$

where N denotes the number of training examples and p the number of parameters (in this case $p_{AR} = 1 + n$ and $p_{ARMA} = 1 + 2n$). Although originally proposed for linear models, this criterion has also been advised for nonlinear estimation (Faraday and Chatfield, 1998).

An *Heuristic* approach (*HEA*) to model selection will be used to draw some preliminary conclusions. Four rules will be used for generating a number of possible models, through time window selection based on the autocorrelation values (Cortez et al., 2001):

A - with all time lags from 1 to a given maximum m: $< 1, 2, ..., m >$ (m was set to 13, a value that was considered sufficient to encompass monthly seasonal and trended effects);

B - with all lags containing autocorrelation values above a given threshold (set to 0.2);

C - with the four lags with highest autocorrelations (in the case of the seasonal trended series, these were taken after differencing, since trend effects may prevail over seasonal ones); and

D - the use of decomposition information; i.e.,

- $< 1, K, K + 1 >$ if the series is seasonal (period K) and trended;
- $< 1, K >$ if the series is seasonal; and
- $< 1 >$ and $< 1, 2 >$ if the series is trended.

The *HEA* procedure is given by the pseudo-code:

```
BEGIN
   Generate models according to rules A, B, C and D
   Estimate parameters of each model, by running the EA
   Compute the BIC for each model
   Select the model with the lowest BIC
END
```

*** insert Table 2 around here ***

As an example, the methodology will be explained in detail for the **prices** *TS* (Table 2). All tests reported in this work were conducted using programming environments developed in *C++* (Neves et al., 1999). The above heuristics were tested in all series of Table 1 using both *EAs* (*AR* and *ARMA*). The results of the last three columns are given in terms of the mean of the thirty runs, being the 95% confidence intervals (Flexer, 1996) also shown for the forecasting error in the test set (column $RMSE_f$).

The lowest training error (column $RMSE_t$) is achieved for the time lags $< 1, 2 >$ and model *ARMA*. The *BIC* criterion works better, by selecting a model with less parameters which provides the best forecast. This behavior occurred consistently in all series, validating the use of this statistic for model selection (Table 3).

*** insert Table 3 around here ***

4. Meta-Evolutionary Algorithm Approach

The *HEA* approach (section 3) only explores a small subset of all possible *AR* and *ARMA* models. Moreover, the time window selection rules are ultimately based on autocorrelation values, which only measure linear interactions that are not adequate for nonlinear series.

An alternative is to use an *EA* optimization procedure to model selection, which is attractive due to its unbiased global search. Typically, *Meta-EAs* (Grenfenstette, 1986), also known as *hierarchical EAs*, are used for the optimization of *EA* parameters (e.g., *population size* or *mutation rate*). In this work, a two-level architecture is proposed, consisting of a meta-level *EA*, used for model selection, and a low-level *EA* (presented in section 2), with parameter estimation purposes.

A binary *Meta-EA* is adopted, where each individual encodes an *ARMA* model, each gene representing a possible coefficient, such that if its value is 1 it exists in the model, otherwise it is not considered (Figure 3). The fitness of each individual in the *Meta-EA* is obtained by decoding its chromosome into the *ARMA* model, running the low-level *EA* to optimize its parameters, and finally calculating the *BIC* value over the training set (Figure 4).

*** insert Figure 3 around here ***

*** insert Figure 4 around here ***

*** insert Table 4 around here ***

The *Meta-EA* works as an optimization procedure of second order, so the tuning of its parameters is not considered crucial. Thus, it was decided to adopt a population size of 50 individuals, with genetic recombination provided by a two point crossover and binary mutation. The maximum *AR* and *MA* orders (P and Q) were set to 13, which sets the binary chromosome size to 27 genes (1 for the constant and 13 for the *AR* and *MA* coefficients). Table 4 shows a synopsis of the relevant parameter values.

The proposed *Meta-EA* was tested on the set of *TS* from Table 1. The best *ARMA* models, obtained by the *Meta-EA* are shown in Table 5. For each *TS*, it is shown the set of *AR* and *MA* coefficients used by the best model, as well as its total number of parameters (column p). The selected models present some differences when compared with the ones given by the *HEA* approach. In general, fewer time lags are used by the *AR* component and the *MA* portion tends to be more considered (it appears in 7 out of 8 series). As expected, the *BIC* values are lower than those provided by the *HEA* strategy, which supports the use of the *Meta-EA*.

*** insert Table 5 around here ***

As an example, the **kobe** forecasts are detailed in Figure 5. In the left side, the mean forecasts of the thirty simulations is plotted against their real values. A *scatterplot* is also shown (in the right), displaying the sorted observations (horizontal x axis) versus the corresponding forecasts (vertical y axis), for each run. Both plots reveal a good fit: in the first case, the two curves are close, while in the second the dots are near the main diagonal (perfect forecast).

*** insert Figure 5 around here ***

5. Overall Comparison

In this section, the obtained results are compared with conventional *TSF* approaches. The *Holt-Winters (HW)* parameters were optimized using a 0.01 grid search for the best *RMSE*, which is a common practice within the forecasting field (Table 6). A different strategy was adopted for the *Box-Jenkins (BJ)* methodology, since the model selection stage is non-trivial, requiring the use of experts. Therefore, it was decided to use a forecasting package (*Forecast Pro*), which includes automatic model selection (Table 7). Although there are known models in the literature (Box and Jenkins, 1976), these were discarded since they only cover four of the tested series. Furthermore, the literature models presented higher forecasting errors.

*** insert Table 6 around here ***

*** insert Table 7 around here ***

Table 8 presents the comparison throughout bio-inspired and conventional methods. The error values in the table are given in terms of the *Theil's U* statistic (Equation 1). This measure takes positive values (the minimum of zero corresponds to a perfect forecast, while the unity equals to a naive *no change* prediction) and makes easier the comparison among the different series and methods.

*** insert Table 8 around here ***

When comparing the evolutionary approaches, the best results are obtained by the *Meta-EA* (the exception is series **deaths**), endorsing the use of the high level *EA* for model selection. Although very simple, *HW* gives a better overall forecasting accuracy on the seasonal series. This is not surprising, since *HW* was developed specifically for these kind of series. However, this scenario differs when considering other series, namely the trended and nonlinear ones, where the *Meta-EA* shows its strength, outperforming both conventional *TSF* methods.

Despite using the same underlying family of *ARMA* models, the *Meta-EA* outperformed the Box-Jenkins methodology in the considered *TSs*. The differences can be explained by the global search provided by the *EAs*, which works at two levels: a better parameter estimation may be achieved in comparison with *least squares*; and a more efficient model selection based on a reliable estimator (*BIC*) in conjunction with a wider search space. This is considered the main contribution of this work, since the *Box-Jenkins* methodology is widely used.

Yet, this effect is achieved with an increase of the computational complexity. For instance, the execution times for generating a **passenger** model on a Intel Pentium IV 1.60 GHz computer were: $20s$ for a *HW* 0.01 grid search; $2.83s$ for a single low-level *EA* run with a full $< 1, 2, \ldots, 13 >$ $ARMA$ optimization; $1h29m18s$ for the best *Meta-EA* solution (obtained in generation 38). However, acceptable solutions can be obtained in earlier stages of the process, as shown in Figure 6, where the *BIC* of the best solution (left) and forecasting error in the test set (right) are plotted against the number of generations elapsed.

6. Conclusions

The surge of new bio-inspired optimization techniques such as *EAs*, has created new exciting possibilities to the field of forecasting. Following such a trend, it is presented in this work a constructive approach to build *TSF* models, assuming no prior knowledge about the behavior of the series (e.g., the use of specific series transformations). Furthermore, the systems that are generated work autonomously and do not require any kind of statistical data analysis.

The main handicap is the computational complexity of the proposed approach. Nevertheless, time complexity could be reduced if a subset of promising models were incorporated into the *EA's* initial population, although this would require the use of a priori information. Since most of the real-world *TS* use daily or monthly data, this is not considered a major concern.

In future work it is intended to enrich the *GA* forecasting models with the integration of nonlinear functions (e.g., logarithmic or trigonometric). Another area of interest may rely on the application of similar techniques to long term and multivariate forecasting. Once the *EAs* revealed good results in parameter optimization and model selection, other optimization meta-heuristics (e.g., *simulated annealing*, *particle swarm optimization* or *ant colony optimization*) can be used in any of the tasks, including hybrid combinations.

References

Agapie, A. and Agapie, A. (1997). Forecasting the Economic Cycles Based on an Extension of the Holt-Winters Model. A Genetic Algorithms Approach. In *Proc. of the IEEE Int. Conf. On Computational Intelligence for Financial Forecasting Engineering (CIFEr'97)*, pages 96–99, New York.

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice.* Oxford University Press.

Box, G. and Jenkins, G. (1976). *Time Series Analysis: Forecasting and Control.* Holden Day, San Francisco, USA.

Cortez, P., Rocha, M., and Neves., J. (2001). Genetic and Evolutionary Algorithms for Time Series Forecasting. In Monostori, L., Váncza, J., and Alis, M., editors, *Engineering of Intelligent Systems: Proc. of IEA/AIE 2001, LNAI 2070*, pages 393–402. Springer.

Faraday, J. and Chatfield, C. (1998). Times Series Forecasting with Neural Networks: A Case Study. *Applied Statistics*, 47:231–250.

Flexer, A. (1996). Statistical Evaluation of Neural Networks Experiments: Minimum Requirements and Current Practice. In *Proc. of the 13th European Meeting on Cybernetics and Systems Research*, volume 2, pages 1005–1008.

Fogel, L. (1999). *Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming.* John Wiley, New York.

Grenfenstette, J. (1986). Optimization of control parameters for Genetic Algorithms. *IEEE Trans. on Systems, Man, and Cybernetics*, 16(1):122–128.

Huang, C. and Yang, H. (1995). A Time Series Approach To Short Term Load Forecasting Through Evolutionary Programming Structures. In *Proc. of the EMPD'95 Int. Conf.*, pages 583–588.

Hyndman, R. (2003). *Time Series Data Library.* Available from http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/.

Makridakis, S., Weelwright, S., and Hyndman, R. (1998). *Forecasting: Methods and Applications*. John Wiley & Sons, New York, USA, third edition.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, USA, Third edition.

Neves, J., Rocha, M., Rodrigues, H., Biscaia, M., and Alves, J. (1999). Adaptive Strategies and the Design of Evolutionary Applications. In W.Banzhaf, J.Daida, A.Eiben, M.Garzon, V.Honavar, M.Jakiela, and R.Smith, editors, *Proc. of the Genetic and Evolutionary Computation Conference (GECCO99)*, pages 473–479. Morgan Kaufmann Publishers.

Rolf, S., Sprave, J., and Urfer, W. (1997). Model Identification and Parameter Estimation of ARMA Models by Means of Evolutionary Algorithms. In *Proc. of IEEE/IAFE Conf. Computational Intelligence for Financial Engineering (CIFEr'97)*, pages 237–243.

Sarle, W. (1995). Stopped Training and Other Remedies for Overfitting. In *Proc. of the 27th Symp. on the Interface of Computer Science and Statistics*, pages 352–360.

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6:461–4.

Table 1. The *Time Series* used in the experiments.

Series	**T***	**Size**	**Range**	**Description**
passengers	ST	144	[104;622]	Monthly airline passengers
paper	ST	120	[215;1006]	Monthly sales of paper (France)
deaths	S	169	[1309;2654]	Monthly deaths/injuries in UK roads
maxtemp	S	240	[12.8;29.5]	Maximum temperature in Melbourne
chemical	T	198	[16.1;18.2]	Chemical concentration readings
prices	T	369	[306;603]	Daily *IBM* stock closing prices
sunspots	N	289	[0.0;190.2]	Annual Wolf's Sunspot Numbers
kobe	N	200	[-10045;11252]	Seismograph of the Kobe earthquake

$\star$ **Type**: ST = Seasonal & Trended, S = Seasonal, T = Trended and N = Nonlinear.

Table 2. Results of the *Heuristic Evolutionary Algorithm* approach applied to the **prices** *TS*.

Model	**Time Lags**	p	**Training**		**Forecasting**
			$RMSE_t$	BIC	$RMSE_f$
	A=B=$< 1, 2, \ldots, 13 >$	14	12.10	1672	11.75±0.84
AR	C=$< 1, 2, 3, 4 >$	5	9.43	1461	8.87±0.40
	$D_{1,2}$=$< 1, 2 >$	3	8.03	1346	7.57±0.09
	D_1=$< 1 >$	2	7.21	**1273**	**7.49**±0.01
	$A = B < 1, 2, \ldots, 13 >$	27	9.57	1495	9.65±0.55
ARMA	C=$< 1, 2, 3, 4 >$	9	7.23	1314	7.83±0.10
	$D_{1,2}$=$< 1, 2 >$	5	**7.16**	1285	7.77±0.02
	D_1=$< 1 >$	3	7.18	1275	7.70±0.02

Table 3. Best forecasting models obtained by the *Heuristic Evolutionary Algorithm* approach to model selection.

Series	**Model**	**Time Lags**	p	BIC	$RMSE_f$
passengers	AR	D=$< 1, 12, 13 >$	4	641	21.9±1.2
paper	AR	D=$< 1, 12, 13 >$	4	777	60.2±2.2
deaths	AR	C=$< 1, 11, 12, 13 >$	5	1440	135.9±1.7
maxtemp	AR	C=$< 1, 11, 12, 13 >$	5	162	0.95±0.02
chemical	$ARMA$	D=$< 1 >$	3	-374	0.36±0.00
prices	AR	D=$< 1 >$	2	1273	7.49±0.00
sunspots	AR	C=$< 1, 2, 9, 10 >$	5	1377	17.9±0.0
kobe	$ARMA$	A=$< 1, 2, \ldots, 13 >$	27	2422	604±36

Table 4. *Meta-Evolutionary Algorithm* and low-level *EA* parameter setup.

	Meta-Level EA	**Low-Level EA**
Encoding	binary	real
Fitness	BIC	$RMSE$
Population size	50	50
Initialization	random {0,1}	random $[-1.0, 1.0]$
Crossover	two-point (80%)	arithmetic (67%)
Mutation	binary (20%)	gaussian perturbation (33%)
Maximum generation	200	1000

Table 5. Best models obtained by the *Meta-Evolutionary Algorithm*.

Series	AR	MA	p	BIC	$RMSE_f$
passengers	<12>	<1,2,3,9,12>	7	563	17.2±0.2
paper	<12>	<>	2	754	52.5±0.1
deaths	<1,11,12>	<13>	4	1420	137±2
maxtemp	<1,7,11,12>	<>	4	161	0.93±0.04
chemical	<1,2>	<1,2,3,4,7,11>	8	-378	0.34±0.00
prices	<1>	<>	1	1273	7.48±0.00
sunspots	<1,2,3,9,10>	<1,9>	8	1362	17.6±0.2
kobe	<1,2,3,7,8,9,13>	<1,3,5,6,12>	12	2219	493±10

Table 6. The *Holt-Winters* forecasting models, given by the parameters α, β and γ from Eq. 3 (obtained by a 0.01 grid search) ; and its forecasting errors.

Series	α	β	γ	K	$RMSE_f$
passengers	0.29	0.03	0.95	12	16.5
paper	0.25	0.01	0.03	12	49.2
deaths	0.36	0.00	0.01	12	135
maxtemp	0.24	0.00	0.11	12	0.72
chemical	0.30	0.00	-	0	0.35
prices	1.00	0.02	-	0	7.54
sunspots	1.00	0.95	-	0	28.3
kobe	0.05	0.00	-	0	3199

Table 7. The forecasting models obtained using the Box-Jenkins methodology, given by the parameters μ, A_i and M_j of Eq. 4 (obtained by the *Forecast Pro* software package); and its forecasting errors.

Series	μ	$A_i^\star$	$M_j^\star$	$RMSE_f$
pass.$^{\dagger\Diamond}$	0.0	$< 1_1, 1_{12}, -1_{13} >$	$< -0.35_1, -0.62_{12}, 0.22_{13} >$	17.8
paper†	0.0	$< 1_1, 1_{12}, -1_{13} >$	$< -0.87_1, -0.80_{12}, 0.70_{13} >$	61.0
deaths†	0.0	$< 1_1, 1_{12}, -1_{13} >$	$< -0.66_1, -0.90_{12}, 0.59_{13} >$	144
maxt.†	0.0	$< 1_1, 1_{12}, -1_{13} >$	$< -0.88_1, -0.89_{12}, 0.78_{13} >$	1.07
chem.$^\Diamond$	0.3	$< 0.90_1 >$	$< -0.56_1 >$	0.36
prices	0.0	$< 1_1 >$	$< 0.12_1 >$	7.72
suns.	14.2	$< 1.39_1, -0.70_2 >$		21.4
kobe	3038	$< 0.71_1, -0.81_2 >$	$< 0.77_1, -0.21_2, -0.06_3 >$	582

$\star$ The values denotes the coeficients (A_i and M_j) and the subscripts denote the time lags (i and j).

$\dagger$ $SARIMA$ models were used for the seasonal series.

$\Diamond$ The data was preprocessed with a natural logarithm transform.

Table 8. Comparison between the different forecasting approaches (based on the $Theil's\ U$ values).

Series	**HW**	**BJ**	**HEA**	**Meta-EA**
passengers	**0.104**	0.118	0.181	0.109
paper	**0.035**	0.076	0.075	0.057
deaths	0.501	0.496	**0.443**	0.448
maxtemp	**0.137**	0.186	0.148	0.143
chemical	0.830	0.861	0.834	**0.777**
prices	1.000	1.006	0.995	**0.994**
sunspots	0.762	0.434	0.305	**0.295**
kobe	0.823	0.027	0.030	**0.020**

List of figure legends:

Figure 1. Autocorrelation coefficients (r_k) of typical *Seasonal & Trended*, *Seasonal*, *Trended* and *Non-Trended* series (x-axis denotes the k time lags).
Figure 2. The eight *TSs* of Table 1 (*passengers*, *paper*, *deaths*, *maxtemp*, *chemical*, *prices*, *sunspots* and *kobe*) in a temporal perspective.

Figure 3. Example of the *Meta-Evolutionary Algorithm* decoding process.

Figure 4. The schematic representation of the *Meta-Evolutionary Algorithm* and the low-level *EA*.

Figure 5. Plots of the *Meta-EA* one-step head **kobe** forecasts and the desired value (left); and the correspondent scatterplot (right).

Figure 6. Plots of the fitness value (left) and forecasting error in the test set (right) of the best solution obtained by the *Meta-EA* in the first 50 generations.

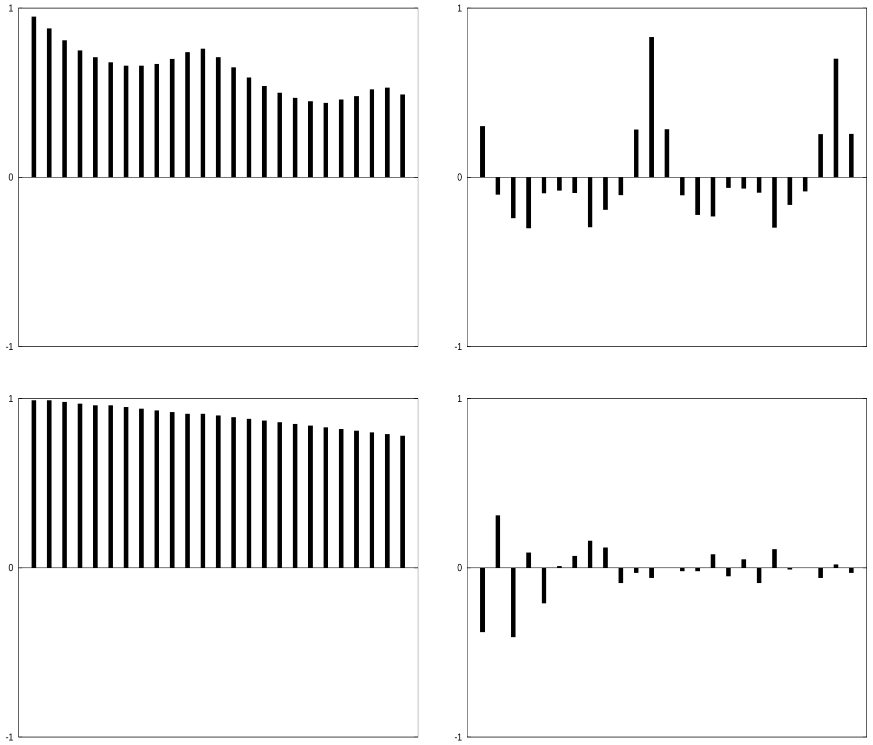

Figure 1.

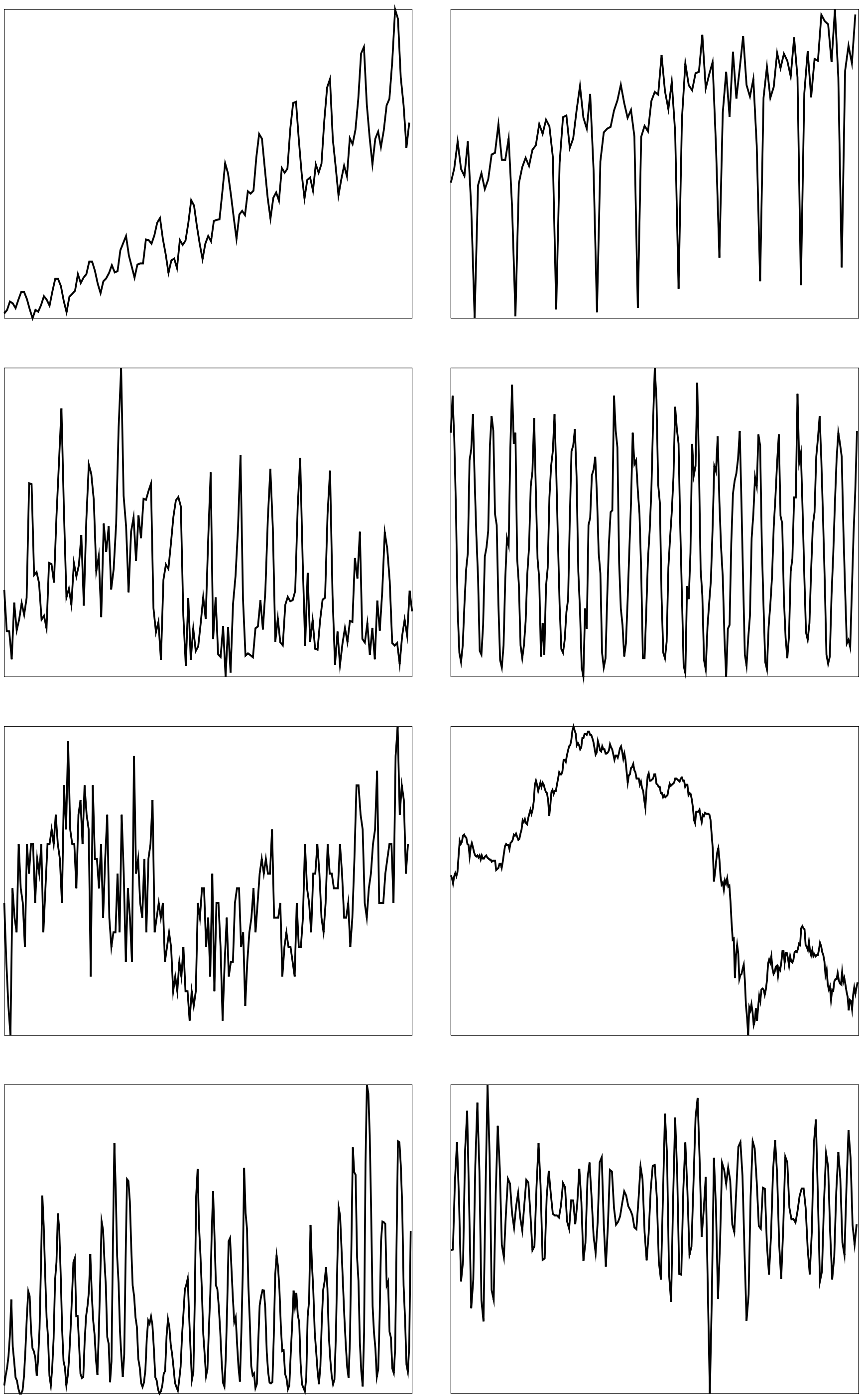

Figure 2.

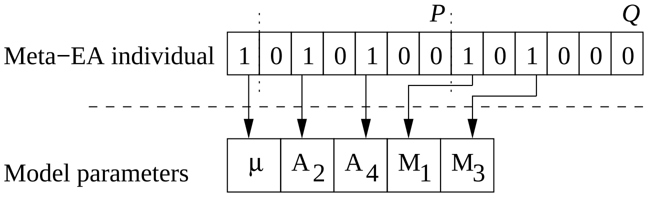

Figure 3.

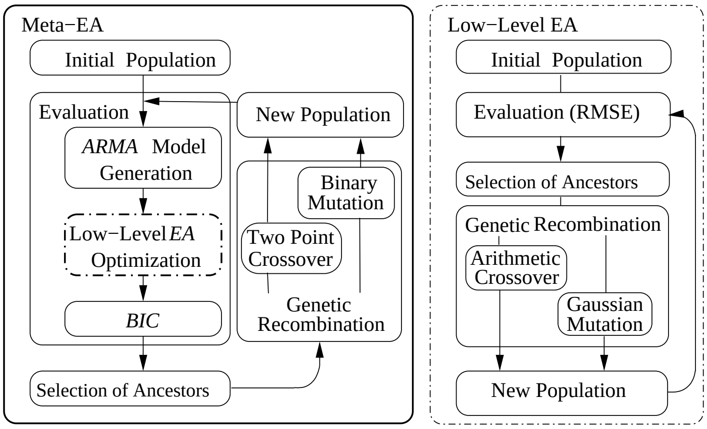

Figure 4.

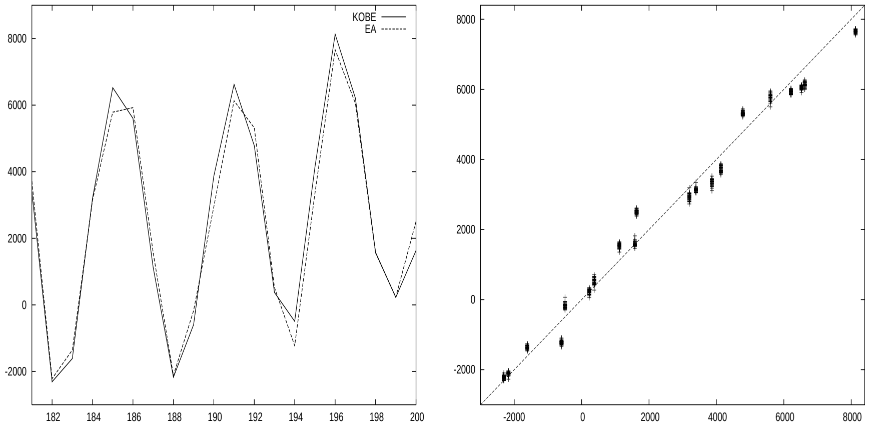

Figure 5.

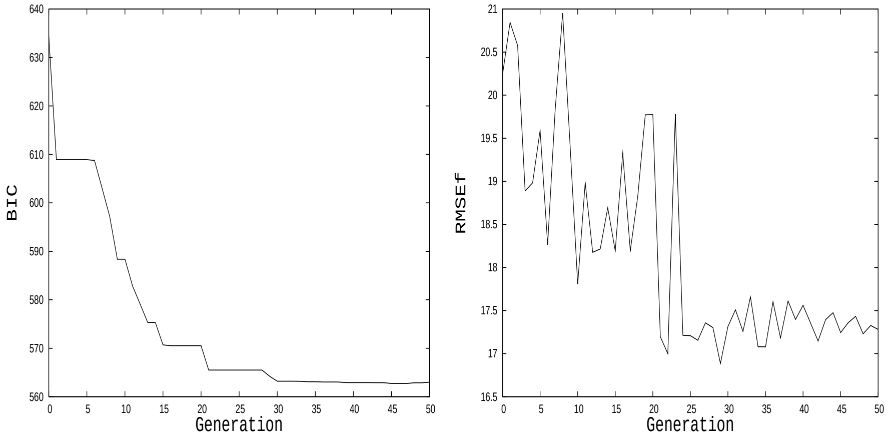
BIC
640
630
620
610
600
590
580
570
560
0
5
10
15
20
25
30
35
40
45
50
Generation
RMSEf
21
20.5
20
19.5
19
18.5
18
17.5
17
16.5
0
5
10
15
20
25
30
35
40
45
50
Generation

Figure 6.