

# A Meta-Genetic Algorithm for Time Series Forecasting

Paulo Cortez<sup>1</sup>, Miguel Rocha<sup>2</sup>, and José Neves<sup>2</sup>

<sup>1</sup> Departamento de Sistemas de Informação - Universidade do Minho  
Campus de Azurém, 4800 Guimarães - PORTUGAL

<sup>2</sup> Departamento de Informática - Universidade do Minho  
Campus de Gualtar, 4710-057 Braga - PORTUGAL  
pcortez@dsi.uminho.pt, mrocha@di.uminho.pt, jneves@di.uminho.pt

**Abstract.** Alternative approaches for *Time Series Forecasting (TSF)* emerged from the Artificial Intelligence arena, where optimization algorithms inspired on natural selection processes, such as *Genetic Algorithms (GAs)* are popular. The present work reports on a two-level architecture, where a (meta-level) binary *GA* will search for the best *TSF* model, being the parameters optimized by a (low-level) *GA*, which encodes real values. The machine's performance of this approach was compared with conventional forecasting methods, exhibiting good results, specially when trended and nonlinear series are considered.

**Keywords:** ARMA Models, (Meta-)Genetic Algorithms, Model Selection, Time Series Forecasting.

## 1 Introduction

*Time Series Forecasting (TSF)*, the forecast of a chronologically ordered variable, corporeals an important problem solving 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*, *Statistics*, and *Computer Science* led to solid *TSF* methods (e.g., *Exponential Smoothing*) that replaced the old fashioned ones, which were primarily based on intuition [10].

An alternative approach for *TSF* arises from the *Artificial Intelligence (AI)* field, where it is observed a trend to look at *Nature* for inspiration, when building problem solving models. In particular, studies on the biological evolution (i.e., derived from living organisms) influenced the loom of powerful tools, such as *Genetic Algorithms (GAs)*, that enriched the potential use of *AI* in a broad set of scientific and engineering arenas, such as the ones of *Combinatorial* and *Numerical Optimization* [11].

*GAs* are innate candidates for parameter estimation, since they implement a global multi-point search, then locating areas of high quality solutions, even when the search space is very large and complex. The use of *GAs* in *TSF* is

motivated by the advantages inherent to the method, such as explicit model representation and adaptive evolutionary search, which prevents them to be locked under undesired local minima. Therefore, it is surprising to realize that the work in applying these techniques to forecasting is so scarce. In fact, although some work had been done in this area, it is not numerous nor noticeable, and it focuses mainly in binary encoding for parameter optimization of conventional methods such as *Holt-Winters* [1] or *ARMA* models [8][4]. However, recent developments in *Genetic Programming (GP)* [2] and *GAs with Real Value Representations (RVRs)* [11], are expected to improve the machine's performance of these techniques.

In previous work [5], *GAs with RVRs* were adopted with some success for *TSF*, being the *GA* used to optimize a set of parameters in a predefined model. The present work aims at the automation of the model selection stage, by means of a meta-level binary *GA*, which will search a space of possible *ARMA* models, which are then optimized by the low-level *GA*.

The paper is organized as follows: firstly, the basic concepts for *TS* analysis, and *GAs* forecasting models are defined; then, a description of the *Meta-GA* and of the conducted experiments is given; finally, the results obtained are presented and compared with other conventional *TSF* methods.

## 2 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). *TS* can arise in a wide set of domains, such as *Finance*, *Production* or *Control*, just to name a few. A *TS* model ( $\hat{x}_t$ ) assumes that past patterns will occur in the future. The *error* of a forecast ( $e_t$ ) is given by the difference between actual values and those predicted by the model:

$$e_t = x_t - \hat{x}_t \quad (1)$$

The overall machine's performance of a model is evaluated by a forecasting accuracy measure, namely the *Sum Squared Error (SSE)*, *Root Mean Squared (RMSE)* and *Normalized Mean Square Error (NMSE)*, given in the form:

$$\begin{aligned} SSE &= \sum_{i=1}^l e_i^2 \\ RMSE &= \sqrt{\frac{SSE}{l}} \\ NMSE &= \frac{SSE}{\sum_{i=1}^l (x_i - \bar{x})^2} \end{aligned} \quad (2)$$

where  $l$  denotes the number of forecasts and  $\bar{x}$  the mean of the *TS*. The other symbols stand for themselves.

A quite successful *TSF* method is *Exponential Smoothing (ES)*, 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* [10].

The *AutoRegressive Integrated Moving-Average (ARIMA)* is another important *TSP* approach to problem solving, going over model identification, parameter estimation, and model validation [3]. 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 *ES*. The global model is based on a linear combination of past values (*AR* components) and errors (*MA* components) This model can be postulated as an *ARMA(P, Q)* one, given in the form:

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

where  $P$  and  $Q$  denote the *AR* and *MA* orders,  $A_i$  and  $M_j$  the *AR* and *MA* coefficients, being  $\mu$  a constant value that enhances the problem solving method's worthiness. Both the constant and the coefficients of the model are estimated using statistical approaches (e.g., least squares methods). The methodology also contemplates the possibility of some kind of transformation in the original data (e.g., logarithmic variation).

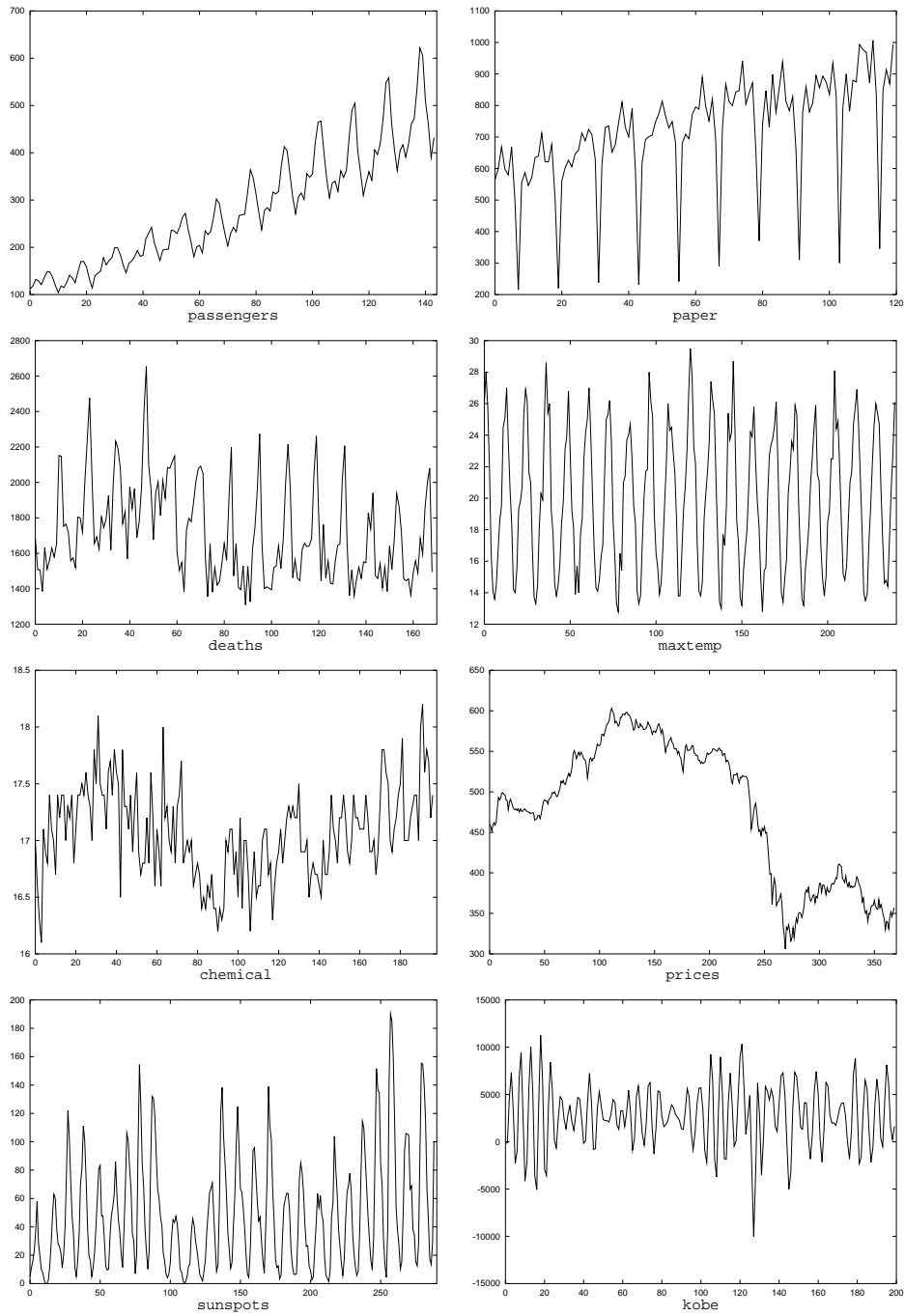
**Table 1.** *Time Series Data*

Series	Type	Domain	Description
passengers	<i>Seasonal</i>	Tourism	Monthly international airline passengers
paper	<i>Trended</i>	Sales	Monthly sales of French paper
deaths	<i>Seasonal</i>	Traffic	Monthly deaths & injuries in UK roads
maxtemp	<i>Seasonal</i>	Meteorology	Maximum temperature in Melbourne
chemical	<i>Trended</i>	Chemical	Chemical concentration readings
prices	<i>Trended</i>	Economy	Daily <i>IBM</i> common stock closing prices
sunspots	<i>Nonlinear</i>	Physics	Annual Wolf's Sunspot Numbers
kobe	<i>Nonlinear</i>	Geology	Seismograph of the Kobe earthquake

To the experiments carried out in this work, a set of eight series were selected (Table 1), ranging from financial markets to natural processes [3][10][9] (Figure 1). All *TSs* were classified into four main categories, that encompass the majority of the *TS's* types, namely *Seasonal and Trended, Seasonal, Trended, Nonlinear*.

### 3 Genetic Algorithm Forecasting Models

The term *Genetic Algorithm (GA)* is used to name a family of computational procedures where an evolving population makes the way to possible solutions to the problem. Each individual codes a solution in a string (*chromosome*) of symbols (*genes*), being assigned a numerical value (*fitness*), that stands for a solution's quality measure. New solutions are created through the application of genetic operators (typically *crossover* and *mutation*). The whole process evolves



**Fig. 1.** The Series of Table 1

via a process of stochastic selection biased to favor individuals with higher fitnesses.

The first *GAs* [7], and most of the ones developed so far, make use of a binary representation; i.e., the solutions to a given problem are coded into a  $\{0, 1\}$  alphabet. However, some authors have argued that when one is faced with problems where the parameters are given by real values, the best strategy is to represent them directly into the chromosome, thus using a *Real-Valued Representation (RVR)*.

In previous work [5], *GAs* with *RVRs* were used to optimize the coefficients of a set of predefined models, which were a sub-class of the general *ARMA*( $P, Q$ ) model, given by the form:

$$\hat{x}_t = g_0 + \sum_{i \in \{1, \dots, P\}} g_i x_{(t-k_i)} + \sum_{j \in \{1, \dots, Q\}} g_{(j+P)} e_{(t-k_j)} \quad (3)$$

where  $g_i$  stands for the  $i$ -th gene of the individuals' chromosome and each component makes use of a sliding time window, denoted by the sequence  $\langle k_1, k_2, \dots, k_n \rangle$ , for a  $n$  order and  $k_i$  time lags.

Under this approach, heuristic strategies, based on *TS* analysis, were used to define the set of attempted models, each encompassing a sub-set of the given coefficients. This strategy implied the comparison of models with different complexities. In fact, complex models tend to correctly handle the training data but fail to generalize, a phenomenon usually termed as *overfitting*.

The usual statistical approach to this situation is *model selection*, where different candidate models are evaluated according to a generalization estimate. Several complex estimators have been developed (e.g., Bootstrapping), which are computationally burdensome [12].

A reasonable alternative is the use of simple statistics that add a penalty to the model that is a function its complexity, such as the *Bayesian Information Criterion (BIC)* [13]:

$$BIC = N \cdot \ln\left(\frac{SSE}{N}\right) + p \cdot \ln(N) \quad (4)$$

where  $N$  denotes the number of training examples and  $p$  the number of model parameters.

With this criterion one was able to, consistently, choose good forecasting models to different kinds of *TSs* (Table 1). In fact, the models with good training errors but with high complexity were prone to overfit. This strategy lead to interesting results, impaired however by the small number of models considered.

## 4 Meta-Genetic Algorithm Approach

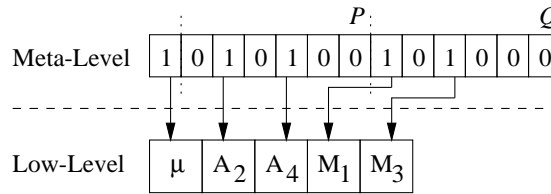
Which is the best *ARMA* model for a given *TS*? The selection of the relevant time lags can improve forecasting (e.g., *ARIMA* models often use the 1, 12 and 13 lags for monthly seasonal trended series). The use of large sliding time windows

can increase the system complexity, diminishing the learning capacity of the model, while small ones may contain insufficient information.

This issue can be addressed by trial-and-error procedures, which tend to be unsuitable due to the huge size of the search spaces involved. A better alternative is to use a *GA* optimization procedure, which is attractive due to its unbiased automated search.

Typically, *Meta-GAs* are used for the optimization of the *GA* parameters (e.g., population size or mutation rate) [6]. In this work, a two-level architecture is proposed, consisting of a meta-level *GA*, used for the model selection task, and a low-level *GA*, with parameter estimation purposes.

A binary *Meta-GA* is adopted, where an individual codes a different *ARMA* model, each gene representing a possible coefficient, such that if its value is 1, then the corresponding coefficient exists in the model, otherwise it is not considered (Figure 2).



**Fig. 2.** Example of a *Meta-GA* Decoding.

This model is optimized by running a *RVR* low-level *GA*, with each gene encoding a coefficient of the *ARMA* model, as given by equation 3. Each individual, is evaluated by the *RMSE* over the training set.

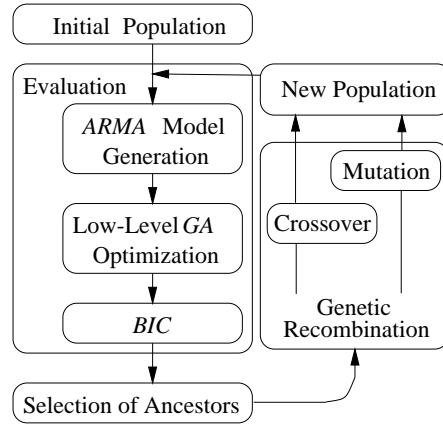
On the other hand, the fitness of each meta-individual is obtained by decoding its chromosome into the low-level *GA*, running it and, finally, calculating the *BIC* value of the optimized model. The overall system is depicted in Figure 3.

## 5 Some Experimental Results

All experiments reported in this work were conducted using programming environments developed in *C++*, under the *Linux* operating system. The proposed *Meta-GA* was tested on the set of *TSs* from Table 1.

The *TSs* were 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. The test set is used to compare the proposed approach with other problem solving methods.

The *Meta-GA* works as an optimization procedure of second order, so the tuning of its parameters is not crucial. Therefore, it was decided to adopt a



**Fig. 3.** The Meta-Evolutionary Approach

population size of 100 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, a value that was considered sufficient to encompass monthly seasonable and trended effects. Thus, the binary chromosome size will be equal to 27 genes (1 for the constant and 13 for the  $AR$  and  $MA$  coefficients).

In both  $GAs$ , the selection procedure is done by converting the fitness value into its ranking on the population and then applying a roulette wheel scheme. In terms of the low-level  $GA$ 's setup, the initial populations' genes were randomly assigned values within the range  $[-1, 1]$ . The population size was set to 50 individuals. In each generation, 40% of the individuals are kept from the original population, and 60% are generated by the application of genetic operators.

The *arithmetic* crossover [11] is responsible for breeding  $\frac{2}{3}$  of the offspring and a *gaussian perturbation* is accountable for the remaining ones. Finally, the  $GA$  is stopped after 1000 epochs. Table 2 shows a synopsis of the relevant parameter values.

The best  $ARMA$  models, obtained by the *Meta-GA* are shown in Table 3. 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  $p$ ). Thirty independent runs of the low-level  $GA$  were performed on the best model, to insure statistical significance, being the forecasting error over the test set presented in terms of the mean and 95% confidence intervals (column  $RMSE_f$ ).

As an example, Figure 4 plots the forecasts for the last 20 elements (10%) of series **kobe**, considering the average of the thirty runs over the optimized  $ARMA$  model. In this case, both curves are close, revealing a good fit by the forecasting model.

**Table 2.** *Meta-GA* Parameter Value Setup.

	Meta-Level GA	Low-Level GA
<b>Encoding</b>	binary	real
<b>Fitness</b>	<i>BIC</i>	<i>RMSE</i>
<b>Population size</b>	100	50
<b>Initialization</b>	random {0,1}	random [-1, 1]
<b>Crossover</b>	two-point (80%)	arithmetic (67%)
<b>Mutation</b>	binary (20%)	gaussian perturbation (33%)
<b>Maximum generation</b>	500	1000

**Table 3.** Best *ARMA* Models Obtained by the *Meta-GA*.

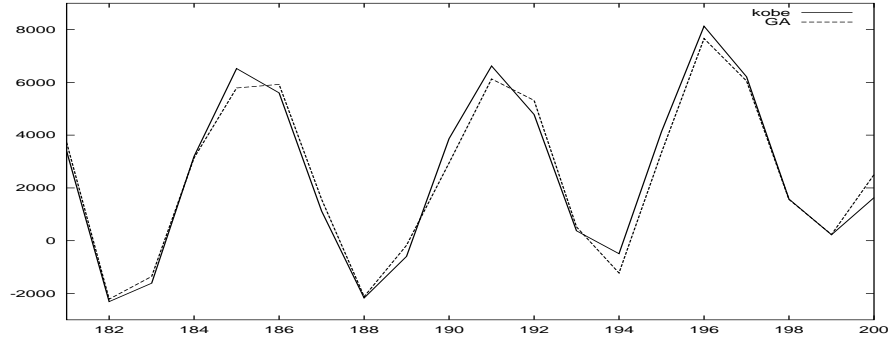
Series	<i>AR</i>	<i>MA</i>	<i>p</i>	<i>RMSE<sub>f</sub></i>
passengers	<12>	<1,2,3,9,12>	7	17.2±0.2
paper	<12>	<>	2	52.5±0.1
deaths	<1,11,12>	<13>	4	137±2
maxtemp	<1,7,11,12>	<>	4	0.93±0.04
chemical	<1,2>	<1,2,3,4,7,11>	8	0.34±0.00
prices	<1>	<>	1	7.48±0.00
sunspots	<1,2,3,9,10>	<1,9>	8	17.6±0.2
kobe	<1,2,3,7,8,9,13>	<1,3,5,6,12>	12	493±10

## 6 Overall Comparison

A comparison throughout bio-inspired and conventional models is given in Table 4. The error values in the table are given in terms of two measures, namely the *RMSE* and the *NMSE* (in brackets). This last measure is included since it makes easier the comparison among the different series and methods. Each model was optimized using all known values from the *TS*, excluding the last 10% values, which will be used for forecasting. The *ES* parameters were optimized using a 0.01 grid search for the best *RMSE*, while the *ARIMA* forecasts were achieved using a forecasting package (*FORECAST PRO*).

*ES* gives a better overall machine's performance on the seasonal series. This is not surprising, since *ES* 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-GA* outperforms both conventional *TSF* methods. Despite using the same underlying general *ARMA* model, the higher flexibility of the *Meta-GA* system allowed it to exceed the machine's performance of the *ARIMA* approach, for all series considered.





**Fig. 4.** One-step Ahead **Kobe** Forecasts.

**Table 4.** Comparison between the Different *TSF* Approaches

Series	ES	ARIMA	Meta-GA
passengers	<b>16.5</b> (0.70%)	17.8 (0.81%)	17.2 (0.75%)
paper	<b>49.2</b> (4.4%)	61.0 (6.8%)	52.5 (5.0%)
deaths	<b>135</b> (37%)	144 (42%)	137 (38%)
maxtemp	<b>0.72</b> (2.5%)	1.07 (5.6%)	0.93 (4.3%)
chemical	0.35 (51%)	0.36 (53%)	<b>0.34</b> (48%)
prices	7.54 (0.39%)	7.72 (0.41%)	<b>7.48</b> (0.38%)
sunspots	28.4 (35%)	21.4 (20%)	<b>17.6</b> (14%)
kobe	3199 (105%)	582 (3.5%)	<b>492</b> (2.5%)

## 7 Conclusions and Future Work

The surge of new bio-inspired optimization techniques such as *GAs*, 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 unique drawback is the increase in the computational effort required.

Comparative experiments among conventional and bio-inspired approaches, with several real series from diverse domains, were held, showing that *ES*, although very simple, presents a good overall machine's performance on seasonal *TS*, being also a method that requires few computational resources. However, when the domain gets more complex, with nonlinear behavior, this kind of methods is clearly not appropriate. The *Meta-GA* approach shows its strength exactly on these scenarios. The results so far obtained prevail both on nonlinear series or on the linear ones, specially on those with trended components.

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.

## Acknowledgments

The work of Paulo Cortez was supported by the Portuguese *Foundation of Science & Technology* through the PRAXIS XXI/BD/13793/97 grant. The work of José Neves was supported by the PRAXIS project PRAXIS/P/EEI/13096/98.

## References

1. Adriana Agapie and Alexandru Agapie. Forecasting the Economic Cycles Based on an Extension of the Holt-Winters Model. A Genetic Algorithms Approach. In *Proceedings of the IEEE Computational Intelligence for Financial Forecasting Engineering*, pages 96–99, 1997.
2. W. Banzhaf, P. Nordin, R. Keller, and F. Francone. *Genetic Programming, An Introduction*. Morgan Kaufmann Publishers, Inc, USA, 1998.
3. G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden Day, San Francisco, USA, 1976.
4. C. Chai, C. Chuek, M. DP, and T. Huat. Time Series Modelling and Forecasting using Genetic Algorithms. In *Proceedings of the First International Conference on Knowledge-Based Intelligent Electronic Systems*, volume 1, pages 260–268, Adelaide, Australia, 1995.
5. P. Cortez, M. Rocha, and J. Neves. Genetic and Evolutionary Algorithms for Time Series Forecasting. In Lázló Monostori, József Váncza, and Moonis Alis, editors, *Engineering of Intelligent Systems: Proceedings of IEA/AIE 2001, LNAI 2070*, pages 393–402, Budapest, Hungary, June 2001. Springer.
6. J. Grenfenstette. Optimization of control parameters for Genetic Algorithms. *IEEE Transaction on Systems, Man, and Cybernetics*, 16(1):122–128, 1986.
7. John Holland. *Adaptation in Natural and Artificial Systems*. PhD thesis, University of Michigan, Ann Arbor, 1975.
8. C. Huang and H. Yang. A Time Series Approach To Short Term Load Forecasting Through Evolutionary Programming Structures. In *Proceedings of the EMPD'95 Internacional Conference*, pages 583–588, 1995.
9. R. Hyndman. *Time Series Data Library*. available from <http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/>, 2001.
10. S. Makridakis and S. Wheelwright. *Forecasting Methods for Management*. John Wiley & Sons, New York, Fifth edition, 1989.
11. Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, USA, Third edition, 1996.
12. W. Sarle. Stopped Training and Other Remedies for Overfitting. In *Proceedings of the 27th Symposium on the Interface of Computer Science and Statistics*, pages 352–360, 1995.
13. G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6:461–4, 1978.