Genetic Search for Threshold Parameters in Time Series Threshold Models: Algorithms and Computer Programs

Roberto Baragona[1] and Domenico Cucina[2]

[1]Department of Sociology and Communication, Sapienza University of Rome, Via Salaria 113 - 00198 Rome, Italy
E-mail: roberto.baragona@uniroma1.it
[2] Department of Statistics, Probability and Applied Statistics, Sapienza University of Rome, Piazzale Aldo Moro 5 - 00100 Rome, Italy
E-mail: c.domenico@email.it

Summary

Threshold models for time series have been often considered to take into account several features of interest. For instance, asymmetric limit cycles, jump phenomena and dependence between amplitude and frequency may be modeled by adapting different structures according to the present state of the process. The most important step for every model identification procedure consists in determining the threshold parameters. Traditional computational methods (e.g. numerical methods for maximizing the likelihood function) cannot solve this problem whilst they may deal with coefficients estimation efficiently given the threshold parameters. These latter need to be determined by means of heuristic methods. Among them the genetic algorithms seem specially promising as they proved their ability at searching in large discrete spaces for a wide class of problems. The main distinctive feature of genetic algorithms is competition among several candidate solutions organized in an algorithmic scheme that uses selection and evolutionary operators to allow the best solution to prevail. We implemented hybrid genetic algorithms to combine heuristic search for threshold parameters with numerical computation of model coefficients. User friendly computer programs are presented to perform the identification and the estimation of some popular threshold models and examples concerned with some real time series are considered.

Keywords: Nonlinear time series, Threshold models, Heteroscedasticity, Genetic algorithms, C++ programming language.

1 Introduction

The practice of time series modeling to represent the dynamic structure of real phenomena has demonstrated that many real time series data cannot be adequately modeled by linear time series models such as the autoregressive moving-average (ARMA) models (Box et al. 1994). Most series have been shown to include a considerable number of nonlinear features such as asymmetric cycles, nonnormality, bimodality, nonreversibility, asymmetric behavior of the variability, volatility clustering, and so on. Some of them have been shown in Tong (1990), Granger & Terasvirta (1993) and Franses & van Dijk (2000). To capture nonlinear features a number of nonlinear time series models have been proposed generally intended as direct generalization of linear structures. Models in such a class yielded successful results concerned with applications in a wide range of fields, e.g. economics and finance, environmental sciences, medicine, digital filters design, engineering and control. In this paper the attention is focused on the class of threshold models characterized by piecewise linear processes separated according to the size of the observed output in comparison with some threshold variables. When each linear regime follows an autoregressive process we have the well known threshold autoregressive (TAR) models proposed by Tong (1978) and later on generalized in a relevant number of contributions. An important TAR model is the self exciting threshold autoregressive (SETAR) model where the model structure changes according to the result of a comparison between a delayed observation and the threshold parameter. Methods for threshold and delay parameters estimation have been proposed by Tsay (1989) and Tong (1990). The SETAR models have been widely employed in the literature to explain various empirical phenomena in an observed time series. See, for example, the work of Chappel et al. (1996) for exchange rate, Tong & Yeung (1991) for beach water pollution, Yadav et al. (1994) for futures markets, Watier & Richardson (1995) for epidemiological applications, Montgomery et al. (1998) for U.S. unemployment, Feucht et al. (1998) for medical studies.

As a natural extension the self exciting threshold autoregressive moving-average (SETARMA) models were proposed by Tong (1990). Properties and characterization of SETARMA models may be found in Brockwell et al. (1992). As a further extension the subset SETARMA (SSETARMA) models may be worth considering (Baragona et al. 2004*a*). Subset models are useful because they allow some model coefficients to be constrained to zero and parsimonious structure to be properly specified.

Further empirical evidence shows that the volatility of a financial or economic time series is often not constant. The autoregressive conditional heteroscedastic (ARCH) models introduced by Engle (1982) constitute established techniques for modeling volatility in economic field and financial markets. The class of ARCH models provides a useful scheme in which the conditional variance of a stationary time series, conditional on its past history, is allowed to vary according to an AR model built on the squared past innovations. There are numerous applications of ARCH-type models in economic and financial time series analysis (see, for instance, Granger & Terasvirta (1993) and Franses & van Dijk (2000)). Bollerslev (1986) extended this formulation to the generalized autoregressive conditional heteroscedastic model (GARCH) where the squared past innovations are modeled according to an ARMA model. A major drawback of the standard ARCH and GARCH models is that the estimated model coefficients are assumed to be fixed throughout the observed period, and they fail to take into account the possibility of asymmetrical regime switching of both average returns and volatility. Tong (1990) suggested to combine the use of a nonlinear model for the conditional mean with a structure for the conditional variance, introducing the so called second generation models. Tong (1990), in particular, has proposed the combination of the SETAR model with the ARCH structure, naming this combination SETAR-ARCH. Li & Lam (1995) assert that the use of this model should address critical issues concerned with the analysis of financial time series, where the asymmetry in the levels can be well captured by a threshold model while the heteroscedasticity can be accounted for by using an ARCH model. In this context, Li & Li (1996) and Liu et al. (1997) combined the features of ARCH and SETAR models and proposed the double threshold autoregressive conditional heteroscedastic (DTARCH) models. Many applications showed that DTARCH models and double threshold generalized ARCH (DTGARCH) models are able to yield better fit and more accurate forecasts of financial time series (see, for instance, Brooks (2001)). Li & Li (1996) derived portmanteau statistics (Q_m, Q_{mm}) based on the residual autocorrelations and squared residual autocorrelations from the DTARCH model for checking the overall goodness of fit.

In both SETARMA and DTGARCH models switching from a regime to another produces an instantaneous change in the model structure. The smooth transition models, for instance the smooth transition autoregressive (STAR) model, allow the model structure to change smoothly according to some transition function (van Dijk et al. 2002). Estimation problems are still difficult, however, because the transition function requires that some parameters have to be suitably determined to fit the data accurately enough. Each parameter in the model is a function of the state of the process. As in the SETARMA and DTGARCH models the state is often a delayed value of the process itself. A flexible approximation that is easier to estimate consists in estimating a broken line close to the curve described by the transition function. This may be done by selecting threshold parameters that define regimes in each of which the approximating function is a straight line. The resulting piecewise function is a continuous function of the state of the process, though not necessarily differentiable. These piecewise linear threshold autoregressive (PLTAR) models have been introduced by Baragona et al. (2004*b*).

Relevant difficulties in building threshold models are to ascertain the number of regimes (at least a reasonable range for it), the threshold parameters and the delay parameter. This problem may be conveniently considered an optimization problem that involves finding the optimum of a criterion function of discrete arguments. If the number of regimes is unknown, and the only available information that may be supplied is concerned with the maximum number of regimes K, the thresholds are to be searched among a large number of alternatives (approximately, this number is $O[n^{(K-1)}]$). If the DT-GARCH model is considered, K and H regimes may be postulated for the average returns and variance respectively. The space of solutions may include $O[n^{(K+H-2)}]$ candidate thresholds. Searching a very large discrete space is a task that has been found convenient to handle by the local search methods, often called general heuristics or meta heuristics (for a review of these methods and applications see, for instance, Pirlot (1996), Bergeret & Besse (1997), Winker (2001) and Winker & Gilli (2004)). Optimization methods in this class are specially useful if, as it is the case here, the objective function has not the usual mathematical properties (continuity, differentiability, concavity) needed for applying gradient based optimization techniques. Among the many available meta heuristic methods the genetic algorithms (GAs) are preferred here because they have desirable properties that make their application very promising in the present context. The GAs were introduced by Holland (1975) to simulate the evolution of either a real or artificial population towards adaptation to the environment. Their potential for functions optimization, however, was early recognized (De Jong 1975). According to this view, the GAs are evolutionary algorithms that process a set of initial solutions through several iterations to find the solution that optimizes the objective function. The solutions that are processed in a given iteration are a small subset of the set that contains all feasible solutions. The main advantages of GA are parallel processing of several feasible solutions and very mild assumptions on the objective function (which is called fitness function (FF)). Appropriate problem-dependent formulations have been developed and described by, for instance, Man et al. (1999), Chen et al. (2001), Haupt & Haupt (2004), Pasia et al. (2005). As far as threshold models are concerned, GAs-based procedure have been suggested by Wu & Chang (2002) for SETAR modeling, Baragona et al. (2004*a*) for SETARMA and subset SETARMA models, Baragona et al. (2004*b*) for PLTAR models, Baragona & Battaglia (2006) and Baragona & Cucina (2008) for DTARCH and DTGARCH models. Convergence properties of GAs have been investigated by De Jong (1975) and Jennison & Sheehan (1995) who offer further insight into the Holland's scheme theorem and, in the Markov chains framework, by Vose (1999) and Reeves & Rowe (2003), among others. Prugel-Bennett & Shapiro (1994) suggested an approach based on statistical mechanics to gain an insight into parameters setting in GAs design.

The GAs-based procedures for threshold models building have been found quite effective through both simulation experiments and applications to real time series. The lack of widespread available and user friendly computer programs is possibly the main reason that explains why GAs-based threshold model building has found little application in practice. We present in this paper some easy-to-use computer programs that perform model identification and estimation of PLTAR, SETAR, DTARCH and DTGARCH models. The procedures have been implemented by a C++ code which operates in a Windows environment. Adaptation to other platforms is straightforward, however. The programs are freely available and may be downloaded from http://w3.uniroma1.it/statstsmeh/index.html. For more information, please contact the authors. Obviously, suggestions and remarks are welcome.

The paper is organized as follows. Next Section is devoted to introduce the classes of threshold models we shall deal with. In Section 3 the GAs-based procedures for threshold models building will be described. In Section 4 examples are illustrated where our computer programs are used to build threshold models for some well known time series that may be considered as benchmarks in the literature. Concluding remarks are drawn in Section 5.

2 Threshold models

Threshold models building procedures are concerned with parameters that identify the model structure (structural parameters) and parameters that define the dependence of the response variable on its lagged values and possibly on exogenous variables (model coefficients). The structural parameters are the delay parameter d, the number of regimes k, the threshold parameters $\mathbf{r} = (r_0, r_1, \dots, r_{k-1}, r_k)'$ (possibly $r_0 = -\infty$ and $r_k = \infty$), the orders (number of coefficients) of autoregressive (AR) and movingaverage (MA) structures, and their corresponding lags. The model coefficients are the AR and MA coefficients and other parameters that are not included into the set of structural parameters. Three models will be considered in detail for which computer programs have been developed. These threshold models are the PLTAR, the SETAR and the DTARCH and DTGARCH. The first two may be viewed as special cases of the general state dependent model (SDM) (Priestley 1988). A SDM for the time series $\{y_t\}$, integer t, takes the form

$$y_t = \sum_{j=1}^{p} \phi_j(z_{t-1}) y_{t-j} + e_t - \sum_{j=1}^{q} \theta_j(z_{t-1}) e_{t-j}, \tag{1}$$

where the array

$$z_{t-1} = (e_{t-q}, \dots, e_{t-1}, y_{t-p}, \dots, y_{t-1})$$

is called the state vector at time $t-1$. In what follows the state variable is assumed y_{t-d}, for given positive integer d.

Let us consider the PLTAR model first. Let the structural parameters be given, then the model coefficients of the SDM model (1) take the special form

$$\begin{aligned} \phi_j(y_{t-d}) &= \alpha_j^{(i)} + \beta_j^{(i)} y_{t-d} \quad \text{if} \quad r_{i-1} < y_{t-d} \le r_i, \qquad i = 1, \ldots, k. \\ \theta_j(y_{t-d}) &= 0, \forall j. \end{aligned} \tag{2}$$

The coefficients $\phi_j(.)$'s are assumed continuous functions of the state y_{t-d}. This constraint is easy to implement and it seems useful as a continuous piecewise linear function may in practice approximate analytic functions to any degree of accuracy. For instance, consider the logistic STAR (LSTAR) model

$$y_t = \sum_{j=1}^{p} \phi_j^{(1)} y_{t-j} + G(y_{t-d}) \sum_{j=1}^{p} \phi_j^{(2)} y_{t-j} + e_t \tag{3}$$

where

$$G(y_{t-d}) = (1 + \exp(-\gamma(y_{t-d} - c)))^{-1}$$

and $\{e_t\}$ is a zero mean white noise with variance σ^2. As an example of comparison between model (2) and model (3) we generated $n = 1050$ observations from the LSTAR model (3) by choosing $\sigma^2 = 1$, $d = 1$, $p = 1$, the AR parameters $\phi_1^{(1)} = -0.4$ and $\phi_1^{(2)} = 0.8$, $\gamma = 1$ and $c = 0.5$. Then, a PLTAR model has been estimated by using a GAs-based procedure. The estimated model includes $k = 3$ regimes in each of which a first order polynomial represents the behavior of the AR parameter as a function of the state variable y_{t-1}. In Figure 1 the varying AR parameter of the LSTAR model (3) is plotted along with the piecewise linear AR parameter of the PLTAR model (1)-(2). The approximation provided by the PLTAR parameter seems adequate to match the logistic curve of the LSTAR parameter. Moreover, 50 out-of-sample one-step-ahead forecasts have been computed from the LSTAR and the PLTAR models. The predicted values are plotted in Figure 2 along with the original time series out-of-sample values. There is evidence that there is a close agreement between forecasts, and both seem able to reproduce fairly well the observations generated from model (3). The mean square errors are 4.6 (LSTAR) and 4.9 (PLTAR). The performance of the two models may be considered similar. The advantage in using the PLTAR model resides in that there is non need to specify in advance any special functional form for the AR parameters.

Let us consider now the SETAR model. Given $k - 1$ threshold parameters $\{r_1 < \ldots < r_{k-1}\}$, and letting $r_0 = -\infty$ and $r_k = +\infty$, then k regimes are identified by the disjoint intervals R_i=$(r_{i-1}, r_i]$, $i = 1, \ldots, k$. A time series $\{y_t\}$ is said to be generated by a SETAR model if

$$y_t = c^{(i)} + \textstyle\sum_{j=1}^{p} \phi_j^{(i)} y_{t-j} + e_t \quad \text{if} \quad r_{i-1} < y_{t-d} \le r_i, \qquad i = 1, \ldots, k, \tag{4}$$

Figure 1: *First order AR parameter for a LSTAR model (solid line) and a PLTAR model (dashed line) plotted against the state variable.*

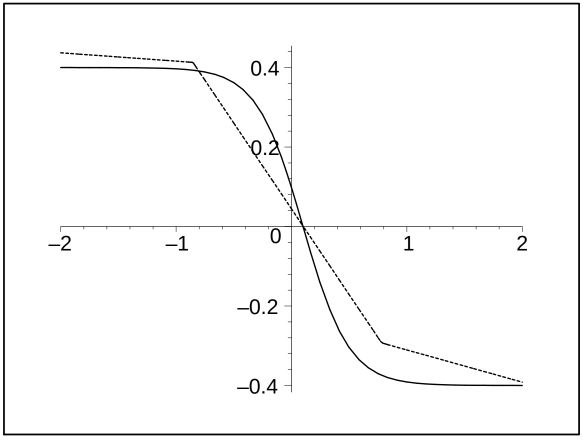
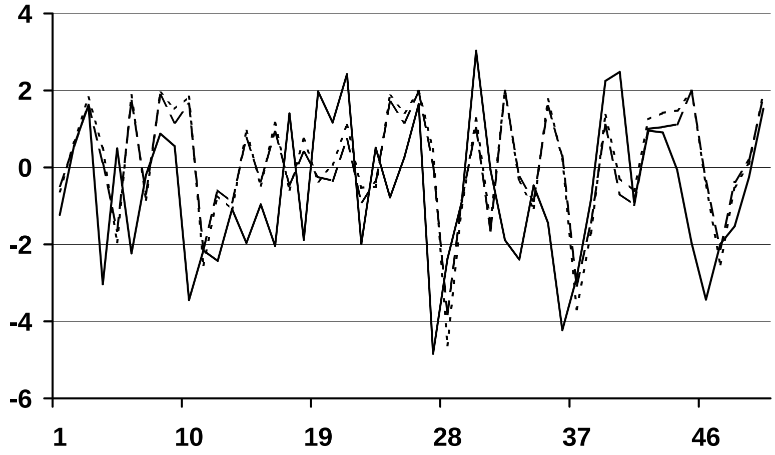


Figure 2: *Out-of-sample one-step-ahead forecasts of a time series generated by a LSTAR model (solid line) yielded by a LSTAR model (dashed line) and a PLTAR model (dotted line).*



where $\{e_t\}$ is a zero mean white noise with constant variance σ^2. Model (4) may be viewed as a special case of the SDM model (1) and of the PLTAR model (2). The AR parameters are taken constant in each regime, and cannot describe any more a continuous line across regimes. In each regime the time series $\{y_t\}$ follows a linear AR model with possibly a constant term added. The orders of these AR models may differ from a regime to another, that is in regime i the AR order is $p^{(i)}$. Equation (4) still holds, however, provided that p is assumed the maximum among the non negative integers $p^{(1)}, \ldots, p^{(k)}$ and $\phi_j^{(i)} = 0$ as soon as $j > p^{(i)}$.

The third class of models we want to consider includes the DTARCH and DTGARCH models. If we allow the variance of the SETAR model (4) to vary, then a convenient way for modeling the innovation variance is to define the conditional variance h_t given the past observations of the time series

$$h_t = \alpha_0 + \sum_{j=1}^{q} \alpha_j e_{t-j}^2. \tag{5}$$

The parameter α_0 has to be greater than 0. The parameters $\alpha_1, \ldots, \alpha_q$ have to be non negative and their sum has to be less than 1. Other assumptions have to be imposed on the α_j's to ensure higher moments to exist. A threshold structure may be postulated for the ARCH model (5) as well. A time series generated by a DTARCH model may be written

$$\begin{array}{rlll} y_t = c^{(i)} + \sum_{j=1}^{p} \phi_j^{(i)} y_{t-j} + e_t & \text{if} & r_{i-1} < y_{t-d} \leq r_i, & i = 1, \ldots, k \\ h_t = \alpha_0^{(i)} + \sum_{j=1}^{q} \alpha_j^{(i)} e_{t-j}^2 & \text{if} & u_{i-1} < e_{t-c} \leq u_i, & i = 1, \ldots, h \end{array} \tag{6}$$

where $\{u_0, u_1, \ldots, u_{h-1}, u_h\}$ are the ARCH threshold parameters, $u_0 = -\infty$ and $u_h = \infty$, h is the number of regimes and c is the delay parameter. More restrictive assumptions are sometimes formulated for the threshold ARCH model, namely that the regimes depend only on the time series $\{y_t\}$. In this case, a change in the regime of y_t implies a contemporaneous change in the regime of h_t.

Likewise, the DTGARCH model may be defined by letting

$$h_t = \alpha_0 + \sum_{j=1}^{s} \beta_j h_{t-j} + \sum_{j=1}^{q} \alpha_j e_{t-j}^2, \tag{7}$$

where the parameters $\beta_1, \ldots, \beta_s$ are non negative. Then we may replace in (6) the structure (5) with (7) to obtain the DTGARCH model

$$\begin{array}{rlll} y_t = c^{(i)} + \sum_{j=1}^{p} \phi_j^{(i)} y_{t-j} + e_t & \text{if} & r_{i-1} < y_{t-d} \leq r_i, & i = 1, \ldots, k \\ h_t = \alpha_0^{(i)} + \sum_{j=1}^{s} \beta_j^{(i)} h_{t-j} + \sum_{j=1}^{q} \alpha_j^{(i)} e_{t-j}^2 & \text{if} & u_{i-1} < e_{t-c} \leq u_i, & i = 1, \ldots, h. \end{array} \tag{8}$$

Ling (1999) showed that sufficient conditions for stationarity of either (6) or (8) are as follows

$$\sum_{j=1}^{p} \max_{i \in (1,k)} |\phi_j^{(i)}| < 1,$$
$$\sum_{j=1}^{q} \max_{i \in (1,h)} \alpha_j^{(i)} + \sum_{j=1}^{s} \max_{i \in (1,h)} \beta_j^{(i)} < 1.$$

However less restrictive assumptions are likely to be needed (Nelson & Cao 1992).

3 Genetic algorithms for threshold models estimation

The GAs that are proposed here address the search for the regimes and threshold parameters. A single vector of threshold parameters has to be determined for PLTAR and SETAR models, while two vectors of threshold parameters have to be determined for DTARCH and DTGARCH models. Basically the algorithm structure remains the same, but searching for thresholds in the first line of (6) or (8) has to proceed in parallel with searching for thresholds in the second line. Let n observations $\{y_1, \ldots, y_n\}$ be available. In case of heteroscedastic models, n innovations may be estimated by fitting a high order AR model to the data. It is convenient to restrict the search for thresholds to the observed data because a finite set may be dealt with. Then the observed data $\{y_1, y_2, \ldots, y_n\}$ have to be arranged in ascending order to obtain the sequence $\{y_{(1)}, y_{(2)}, \ldots, y_{(n)}\}$. There is no loss of generality (in sample) as $y_{t-d} \leq r_i \Longleftrightarrow y_{t-d} \leq y_{t(j)}$ where $y_{t(j)}$ is the greatest observation less than or equal to r_i. A minimum number m of observations in each regime has to be chosen to ensure that each linear model may be estimated safely. So we may consider only the subsequence $Y = \{y_{(m+1)}, y_{(m+2)}, \ldots, y_{(n-m)}\}$. As a matter of fact, the first and last m observations in the ascending sequence may be dropped because at least m observations in each regime are required. In case of DTARCH and DTGARCH models, where independent changes in regimes for y_t and h_t are allowed to occur, the thresholds u_i have to be searched for in the subsequence $E = \{e_{(m+1)}, e_{(m+2)}, \ldots, e_{(n-m)}\}$ of the innovations arranged in ascending order. The requirement that at least m observations have to belong to each regime obviously imposes a constraint on the number of regimes. It is more convenient, however, to define explicitly a maximum number K of regimes, so that $1 \leq k \leq K$. This simplifies somewhat the GA implementation. A similar constraint applies for the independent thresholds for h_t in heteroscedastic models, that is $1 \leq h \leq H$. As the GA searches for the optimal sequence of thresholds, each and every solution is a sequence of real numbers $\{r_1, \ldots, r_{k-1}\}$ where $r_i \in Y$ and $r_i \neq r_j$ if $i \neq j$

$(i, j = 1, \ldots, k-1)$. For heteroscedastic models with independent changes in regime of y_t and h_t a solution is a sequence $\{r_1, \ldots, r_{k-1}; u_1, \ldots, u_{h-1}\}$ where $r_i \in Y$ and $u_i \in E$. The additional requirement $r_i \neq r_j$ and $u_i \neq u_j$ applies. Furthermore, given the maximum number of regimes K (or K and H), the constraint $k \leq K$ ($k \leq K$ and $h \leq H$) is required.

In the present implementation the GAs operators do not manipulate a thresholds sequence directly but this latter has to be encoded in a string of characters. In the GAs framework each string is called a *chromosome* and each character is called a *gene*. A chromosome may be written as a vector of genes. Let ℓ denote the length of the chromosome, then

$$\mathbf{w} = (w_1, w_2, \ldots, w_\ell)'$$

represents a potential solution that has to be decoded to yield the corresponding sequence of thresholds $\{r_1, \ldots, r_{k-1}\}$. The mapping $\mathbf{w} \leftrightarrow \{r_1, \ldots, r_{k-1}\}$ allows the algorithm to represent a solution either as a chromosome or a threshold sequence.

The GAs assume as a search space the set W of all vectors $\mathbf{w}$ (or $\mathbf{w}^* = [\mathbf{w}; \mathbf{v}]$)) each of which encodes (and may be decoded from) an element of the set Y ($Y \cup E$). Only a subset $P \subset W$ is processed in an iteration of the algorithm. The set of s vector $P = \{\mathbf{w}^{(i)}, i = 1, \ldots, s\}$ is called a *population*, though P usually is only a small subset of W, and s is the *population size.* The GA starts with an initial population, then proceeds through a pre-specified number of iterations N. The population P in a GA iteration is called sometimes a *generation.* In each iteration the vectors in the current population are evaluated by means of a pre-specified objective function, called *fitness function* (FF). The FF computation for the ith chromosome involves usually the estimation of the model (PLTAR, SETAR, DTARCH or DTGARCH) by assuming the number of regimes and threshold parameters as encoded in $\mathbf{w}^{(i)}$. In the present paper (and in our computer programs) the FF evaluation is done as follows. The AR order is found by trying several values p in a pre-specified range $[1, P]$ and choosing the one which minimizes the AIC criterion (Akaike 1977)

$$\text{AIC} = -2\{\log - \text{likelihood}\} + 2\{\text{numbers of parameters}\}.$$

Moreover, the estimation is performed for each values of the delay parameter d in the range $[1, p]$. Computations are quite demanding, so it is important that the space of solutions be explored efficiently. The GA design aims at optimize the searching procedure. Once the model is identified and estimated completely then the overall AIC is computed (one of its several variants (Bozdogan & Bearse 2003) may be used as well). The FF is computed for each vector $\mathbf{w}^{(i)}$ and its value f_i is associated to $\mathbf{w}^{(i)}$. Note that the FF has to be positive and as greater as better the potential solution $\mathbf{w}^{(i)}$, so the AIC has to be transformed to meet these requirements. For instance, we set FF $=\exp(-\mathrm{AIC}/c)$, where c is a constant which is introduced to avoid overflows and to scale the FF suitably.

Here two different encodings are implemented.

3.1 Code 1

Let $\ell = n-2m$ and $\mathbf{w}$ be a binary vector of length ℓ, and define the mapping $w_i \leftrightarrow y_{(i+m)}, i = 1, \ldots, \ell$. If $w_i = 1$, then $y_{(i+m)}$ is a threshold parameter (i.e. $r_{i_\alpha} = y_{(i+m)}$), and $w_i = 0$ otherwise. The number of regimes is given by $w_1 + w_2 \ldots + w_\ell + 1$. In case of heteroscedastic models another vector $\mathbf{v}$ has to be introduced that parallels the subsequence E. This encoding is straightforward but for long time series the length of the chromosome may become very large. In addition, the number of ones in the binary string $\mathbf{w}$ has to be checked at each step of the algorithm because of the constraint $k \leq K$ ($k \leq K$ and $h \leq H$) and the requirement that in each regime there have to be more than m observations. If these constraints are not fulfilled a legalization step is needed which may involve either the heuristic adjustment of the chromosome or the introduction of a penalization or discarding the illegal potential solution. In any case additional computations are needed.

3.2 Code 2

Each chromosome $\mathbf{w}$ is composed of $\ell = K-1$ positive integer numbers $(g_1, g_2, \ldots, g_{K-1})$, where $g_i \in (m, n)$, specifying the number of observations that belong to each regime. The mapping $w \leftrightarrow \{r_1, \ldots, r_{k-1}\}$ ($w \leftrightarrow \{r_1, \ldots, r_{k-1}; u_1, \ldots, u_{h-1}\}$) is defined as follows. If $g_1 \leq n-m$, then we have a threshold model and the first regime has g_1 observations in the interval $[y_{(1)}, y_{(g_1)}]$. The first threshold is assumed $r_1 = y_{(g_1)}$. If $g_1 > n-m$ a single regime model has to be estimated and the decoding procedure ends. If this latter is not the case, the second regime has g_2 observations in the interval $[r_1, y_{(g_1+g_2)}$, the second threshold is $r_2 = y_{(g_1+g_2)}$, and so on. The last regime includes the observations larger than $y_{(g_1+\ldots+g_{k-1})}$ where

$$k-1 = \arg \begin{array}{c}\max\\ k \in [1,K]\end{array} \{g_1 + \ldots + g_{k-1} < n-m\}.$$

The number of regimes k is implicitly determined by the genes g_j and $0 \leq k \leq K$. Though less easy to interpret, this encoding allows legal chromosomes only (i.e., corresponding to models which satisfy the requirements concerned with the maximum number of regimes K and the minimum number of observations m in each regime).

3.3 Initialization, selection and genetic operators

Given the population size s, a set of vectors $\{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(s)}\}$ has to be chosen, usually at random. The population in each iteration is processed by means of the usual genetic operators selection, crossover and mutation. We try to adopt the most commonly employed GAs operators whenever possible, but some special features are introduced to exploit the peculiarity of the present problem. A description of selection and of the two operators crossover and mutation follows.

Selection. The first choice is the well known roulette wheel rule. Available alternatives are, for instance, the stochastic universal sampling or tournament selection (Goldberg 1989). According to the roulette wheel selection, each vector in the current population is assigned a probability to be selected for the next generation proportional to its FF. Any vector in the population may be selected, even more than once, and placed in the population that will be processed in the next iteration. It may happen, however, that a vector is not selected, that is it is discarded. To avoid discarding good solutions, the elitist strategy is adopted to amend the pure stochastic selection. A vector that has been discarded is reinserted in the new population if its FF is greater than the FF of each of the selected vectors. In this case, this best vector replaces the vector that in the new population has the worst FF. This way the size s of the population remains unchanged through the iterations. Note that holding s fixed is a convenient choice only and variable size generations are manageable. Also, selection may be performed only on a fraction of the chromosomes in the current population. This fraction G of chromosomes that are to be replaced by new chromosomes in the next generation is called generational gap.

Crossover. It is advisable to resort to different implementations of crossover whether the *code* 1 or *code* 2 is used. However, in both cases a probability of crossover p_c has to be pre-specified. Then, in the current population, $[p_c s/2]$ vector pairs are selected at random. A vector may be selected more than once to enter a pair. The selected pairs are processed one at the time, and both vectors are updated soon after each crossover.

For the *code* 1 we do not recommend the common crossover operators and the following special procedure may be outlined. Let $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$ denote the two vectors in the pair. All bits of the two vectors are examined, and the indexes where the first one, or the second one, or both, have bit value equal to 1 are recorded. If there are no bits equal to 1 the crossover does not take place. Otherwise, let $\{i_1, \ldots, i_\mu\}$ denote the indexes of the bits equal to 1. An index is selected at random in this set, and let us denote the selected bit index i_α. Then, two cases are to be considered: (1) Either $w_{i_\alpha}^{(1)} = 1$ or $w_{i_\alpha}^{(2)} = 1$, but not both. If, for instance, $w_{i_\alpha}^{(1)} = 0$ and $w_{i_\alpha}^{(2)} = 1$, then we set $w_{i_\alpha}^{(1)} = 1$ and $w_{i_\alpha}^{(2)} = 0$. The number of regimes encoded in the second vector decreases by one, whilst a new regime adds to the number of regimes encoded in the first vector. (2) Both $w_{i_\alpha}^{(1)} = 1$ and $w_{i_\alpha}^{(2)} = 1$. In this case, the crossover does not take place. Note that the crossover may yield only one change of regime at most in each of the two vectors, either adding one, or deleting one. The proposed implementation seems convenient to avoid excessive change in the vectors that undergo the crossover. As a matter of fact, it obeys to a general rule that the vectors yielded by the crossover do not have to differ completely from the original vectors. The usual one-point crossover, for instance, has been often found to upset the vectors structure.

For the *code* 2 we may consider the common crossover operators suited to the present framework as this encoding yields only legal chromosomes (i.e., corresponding to models which satisfy the requirements related to number of regimes and number of observations in each regime). So the crossover operator may be applied as usual (for instance, we adopted the two-point crossover).

Mutation. A probability of mutation p_m, usually quite small, less than 0.1, has to be pre-specified. A vector in the current population undergoes mutation with probability p_m. As for crossover, for mutation too two different implementations have to be designed to fit the different encodings.

As far as the *code* 1 is concerned, let mutation occur for the vector $\mathbf{w}^{(i)}$. A bit of this vector is chosen at random, $w_j^{(i)}$, say, and flips, that is $w_j^{(i)}$ is replaced by $1 - w_j^{(i)}$. If $w_j^{(i)} = 0$ then, after mutation, $w_j^{(i)} = 1$ and a new regime is added. If otherwise $w_j^{(i)} = 1$ then, after mutation, $w_j^{(i)} = 0$ and an existing regime is deleted. If we adopted the usual mutation operator, any bit of any vector would be allowed to flip with probability p_m. Due to the constraint on the number of regimes, this mutation operator is likely to yield new regimes while it is unlikely to delete some regimes. As a vector with more than $K - 1$ bits equal to 1 has to be discarded, the usual mutation in most cases would be an useless operator.

For *code* 2 we suggest a special mutation operator as well. We may consider that mutation has to modify a gene by the smallest possible amount. Our choice is that, if the gene g mutates, the new value is chosen with uniform probability in the interval $(\max\{m, g - m/2\}, g + m/2)$.

3.4 Statistical inference on parameters

For the model coefficients asymptotic distribution or at least asymptotic standard errors are provided by the numerical optimization techniques used for estimation. The structural parameters or all parameters that are estimated by using the GAs in non-hybrid form or by heuristic search need special techniques to assess inferential procedures. Such parameters are the number of regimes (k, h) and the threshold parameters $\{r_1, \ldots, r_{k-1}\}$ and $\{u_1, \ldots, u_{h-1}\}$, the delay parameters (d, c), the linear model orders (p, q, s). The GAs aim at producing point estimates of parameters primarily. However, in statistical model building and specially in empirical applications it is of chief importance that the estimation algorithm could provide the user with standard errors of the estimates, p-values or other statistics useful to make statistical inference on parameters. For instance, in Chatterjee & Laudato (1997) the GAs have been proposed for use in statistical inference procedures. Several devices may be used to obtain both estimates and standard errors of the estimates. The most common suggestion consists in using bootstrap concepts to allow confidence intervals to be computed, that is the bootstrap may be used for re-sampling the entire population and estimating the parameters from the re-sampled data. There are simpler alternatives though not as reliable as the bootstrap and mainly motivated by saving computation resources specially those that exploit the information provided by a single run of the GA.

- The GA may run several times in each of which the random numbers stream changes.
- The final GA population may be considered as a sample from which the empirical distribution of the estimates may be computed.
- According to the interpretation of GAs as Markov chains the distribution of the estimates may be computed from the best fit chromosomes recorded in the latest iterations of the GA.

These procedures deserve further investigation and development and some will be included in the revised version of our computer programs.

4 Examples of applications

The Canadian lynx data, the Hong Kong Hang Seng index (HSI) and French Franc/Deutschmark exchange rate are well known time series that may serve as a benchmark to illustrate our GA based procedures. The software that is used is written in C++ and is implemented so as to result easy to use for fitting threshold models to the data. Downloadable programs and more information are available on the web site http://w3.uniroma1.it/statstsmeh/index.html.

4.1 Fitting SETAR models to the French Franc/Deutschmark exchange rate

Let us consider the French Franc/Deutschmark exchange rate over period 1 May 1990 to 30 March 1992 first. As in Chappel et al. (1996) and Brooks (2001) the models are estimated in the natural logarithms of levels rather than the first differences. For comparing our results with that of Chappel et al. (1996) the first $n = 450$ observations are used for estimating the model, while further 50 observations are used for checking the forecasting performance of the fitted model. The following choices have been made to run the program. As regards the GAs parameters, the population size has been set $s = 50$, the number of iterations $N = 300$, the generational gap $G = 1$, the probability of crossover $p_c = 0.9$, the probability of mutation $p_m = 0.001$, the fitness function $\mathrm{FF} = \exp(-\mathrm{AIC}/c)$ with $c = 1$ and the elitist strategy has been employed within the selection procedure. For the model parameters we set the maximum order of all linear models equal to 4, the values of delay parameter $d = 1, 2, 3$, the maximum number of regimes $K = 4$ and the minimum number of observations per regime $m = 60$.

Our best estimated SETAR models and the SETAR models fitted by Chappel et al. (1996) are reported in Table 1, together with the AIC values, the portmanteau statistics (Q) (Ljung & Box 1978) and the mean square errors (MSEs) computed on one-step-ahead forecasts. In these tables models I and II are SETAR models with two and three thresholds respectively estimated by our program while models III and IV are taken from Chappel et al. (1996) (pp. 160–161). These results show that the performance of our SETAR models is slightly better.

Table 1: *Results for the French Franc/Deutschmark exchange rate*

	Model I	Model II	Model III	Model IV
Model	$(2; 1, 3)$	$(3; 1, 1, 3)$	$(2; 1, 3)$	$(3; 1, 1, 3)$
Threshold	5.8308	5.8183; 5.8292	5.8306	5.819; 5.8306
Observation	375; 72	116; 202; 129	344; 103	119; 225; 103
Delay	1	1	1	1
AIC	−14.323	−14.294	−14.291	−14.314
Q	4.302	3.103	6.482	6.643
MSE	1.76×10^{-7}	1.73×10^{-7}	1.80×10^{-7}	1.80×10^{-7}

We may observe that we fit a SETAR model if either there is prior knowledge that this model is appropriate or comparison is made among many alternative models. Otherwise an alternative choice consists in fitting a PLTAR model that may approximate a wide range of models.

4.2 Fitting a PLTAR model to the Canadian lynx data

The PLTAR model is to be recommended when no prior knowledge exists about the functional form of the appropriate model. As an application, a PLTAR model has been estimated for the Canadian lynx data. This time series consists of the annual records of the number of lynx trapped in the Mckenzie River district of North-west Canada from 1821 to 1934. The number of observations is $n = 114$. This time series has been extensively studied (see, for instance, Tong (1990) pp. 380–381, Table 7.5). The data are transformed as $\log_{10}$(number recorded as trapped in year 1820+t), $t = 1, \ldots, 114$. As far as the GAs parameters are concerned, we set the population size $s = 30$, the number of iterations $N = 300$, the generational gap $G = 1$, the probability of crossover $p_c = 0.8$, the probability of mutation $p_m = 0.001$, the fitness function $\mathrm{FF} = \exp(-\mathrm{AIC}/c)$ with $c = 1$ and selection with elitist strategy. For the model parameters we set the maximum order of all linear models equal to 4, the values of delay parameter $d = 1, 2, 3$, the maximum number of regimes $K = 4$ and the minimum number of observations per regime $m = 25$. We have as best model estimated by our GAs-based procedure

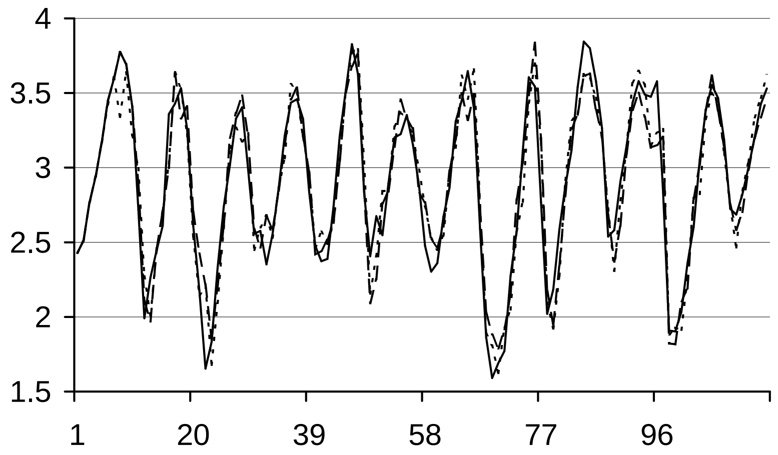
$$y_t = \begin{cases} .3230 + (1.2106 - .1037 y_{t-2}) y_{t-1} + (-.7704 + .4257 y_{t-2}) y_{t-2} \\ +(2.0506 - .8498 y_{t-2}) y_{t-3} + (-1.1132 + .3726 y_{t-2}) y_{t-4} + e_t \\ \qquad \text{if } \; y_{t-2} \leq 3.3101, \\ .0484 + (-9.8474 + 3.2370 y_{t-2}) y_{t-1} + (25.4669 - 7.5009 y_{t-2}) y_{t-2} \\ +(-24.9680 + 7.3127 y_{t-2}) y_{t-3} + (10.1568 - 3.0322 y_{t-2}) y_{t-4} + e_t \\ \qquad \text{if } \; y_{t-2} > 3.3101. \end{cases}$$

Note that the number of coefficients is twice the AR order due to the particular structure of the PLTAR model. Parameters are considered in pairs, that is the constant and the slope of the linear function of the state observation count as a single parameter. The PLTAR models seem able to fit the observed time series at high degree of accuracy. In Figure 3 the observed time series is plotted along with the one-step-ahead forecasts computed by the SETAR model as estimated in Tong (1990) (p. 387) and those computed by our PLTAR model. The two prediction lines seem to overlap almost always. For the PLTAR model we have MSE 0.0121, 18 coefficients and the AIC is equal to -273.78. For comparison, for the SETAR model reported in Tong (1990) the MSE equals 0.0136 and the AIC, with 9 coefficients, is equal to -270.0.

4.3 Fitting DTARCH and DTGARCH models

We illustrate now the DTARCH modeling procedure using the $n = 260$ observations of the Hong Kong Hang Seng index (HSI) data collected in year 1988 (Li & Li 1996). Since the time series is non stationary we transform

Figure 3: *Plot of the Canadian lynx data (solid line) and the one-step-ahead forecasts computed on the SETAR model reported in Tong (1990) (dashed line) and the PLTAR model (dotted line).*



the series as the log-difference of the index. The following choices have been made to run the program. As regards the GAs parameters, the population size has been set $s = 30$, the number of iterations $N = 300$, the generational gap $G = 1$, the probability of crossover $p_c = 0.8$, the probability of mutation $p_m = 0.001$, the fitness function $\mathrm{FF} = \exp(-\mathrm{AIC}/c)$ with $c = 1$ and the elitist strategy has been employed within the selection procedure. For the model parameters we set the maximum order of all linear models equal to 4, the values of delay parameter $d = 1, 2, 3$, the maximum number of regimes $K = 3$ and the minimum number of observations per regime $m = 30$. The return series determines the change in the regime of both returns and volatility models. The overall estimated model is as follows.

$$y_t = \begin{cases} -0.000576 & +0.003075 y_{t-1} & +e_t & \text{if} & y_{t-1} \leq 0.002090, \\ -0.112925 & -0.132312 y_{t-1} & +e_t & \text{if} & y_{t-1} > 0.002090, \end{cases}$$

$$h_t = \begin{cases} 0.000014 & +0.150554 e_{t-1}^2 & \text{if} & y_{t-1} \leq 0.002090, \\ 0.000017 & +0.176816 e_{t-1}^2 & \text{if} & y_{t-1} > 0.002090. \end{cases}$$

The diagnostics Q_m and Q_{mm} (Li & Li 1996, Liu et al. 1997) suggest adequacy of the fitted models for the conditional mean and variance respectively. The AIC is equal to -1277.922. For comparison, the AIC computed for the model reported in Li & Li (1996) is equal to -1244.732.

A DTGARCH model has been fitted to the French Franc/Deutschmark exchange rate time series data set using a GAs based procedure by Baragona & Cucina (2008). Results therein reported are consistent with those reported in Brooks (2001) for the three-regimes model. In both cases the change in the regime depends on the return series only. There are several differences in the identification and estimation procedures, as in Brooks (2001) the delay parameter ($d = 1$) and the number of regimes (either $k = 2$ or $k = 3$) have been chosen on theoretical grounds while Baragona & Cucina (2008) used the GAs-based procedure to find the number of regimes ($k = 3$) and each proposal model has been tried with $d = 1, 2, 3$ and the one corresponding to the minimum AIC has been chosen. The two algorithms yielded the same value for the delay parameter ($d = 1$). For the three-regimes models the thresholds provided by the GAs-based procedure, $r_1 = 5.8159$ and $r_2 = 5.829$, are close to those provided in Brooks (2001) and in agreement with those displayed in Table 1 for Models II and IV. As far as model orders and coefficients are concerned there are differences between the models estimated by Baragona & Cucina (2008) and Brooks (2001). First, Brooks (2001) fitted GARCH(1,1) models to the data in each regime while model orders in Baragona & Cucina (2008) have been chosen according to the minimum AIC criterion. Second, Brooks (2001) used the BFGS optimization algorithm while Baragona & Cucina (2008) used the Powell's method. The diagnostic checking reported in Baragona & Cucina (2008) uses the test statistics Q_m (for the conditional mean) and Q_{mm} (for the conditional variance) and supports adequacy of the estimated DTGARCH model. The GAs-based program for identifying and estimating a DTGARCH model is available on the web site http://w3.uniroma1.it/statstsmeh/index.html as well.

5 Conclusions

Using GAs for identifying and estimating threshold models in time series seems promising according to Monte Carlo simulation results reported in the literature. Yet widespread applications are rather limited mainly because of some lack of easy-to-use computer programs. In this paper we present GAs-based procedures for handling some popular threshold models. The computer programs that implement these GAs-based methods are presented are available on the web site http://w3.uniroma1.it/statstsmeh/index.html. Programs are flexible, in that allow sensible parameters to be specified by the experienced users, and user-friendly, as default parameters are provided that proved to work well in a wide range of applications.

References

Akaike, H. (1977), An entropy maximisation principle, *in* P. R. Krishnaiah, ed., 'Proceedings of the Symposium on Applied Statistics', North-Holland, Amsterdam, pp. 27–41.

Baragona, R. & Battaglia, F. (2006), Genetic algorithms for building double threshold generalized autoregressive conditional heteroscedastic models of time series, *in* A. Rizzi & M. Vichi, eds, 'Proceedings in Computational Statistics', COMPSTAT 2006, Rome, Italy, pp. 441–452.

Baragona, R., Battaglia, F. & Cucina, D. (2004*a*), 'Estimating threshold subset autoregressive moving-average models by genetic algorithms', *Metron* **62**, 39–61.

Baragona, R., Battaglia, F. & Cucina, D. (2004*b*), 'Fitting piecewise linear threshold autoregressive models by means of genetic algorithms', *Computational Statistics & Data Analysis* **47**, 277–295.

Baragona, R. & Cucina, D. (2008), 'Double threshold autoregressive conditionally heteroscedastic model building by genetic algorithms', *Journal of Statistical Computation and Simulation* **78**, 541–558.

Bergeret, F. & Besse, P. (1997), 'Simulated annealing, weighted simulated annealing and genetic algorithms at work', *Computational statistics* **12**, 431–445.

Bollerslev, T. (1986), 'A generalized autoregressive conditional heteroskedasticity', *Journal of Econometrics* **31**, 307–327.

Box, G. E. P., Jenkins, G. M. & Reinsel, G. C. (1994), *Time Series Analysis. Forecasting and Control (3rd edition)*, Prentice Hall, Englewood Cliffs, New Jersey.

Bozdogan, H. & Bearse, P. (2003), Icomp: A new model-selection criterion, *in* H. H. Bock, ed., 'Classification and Related Methods of Data Analysis', Elsevier Science Publishers (North Holland), Amsterdam, pp. 599–608.

Brockwell, P., Liu, J. & Tweedie, R. (1992), 'On the existence of stationary threshold autoregressive moving-average processes', *Journal of Time Series Analysis* **13**, 95–107.

Brooks, C. (2001), 'A double-threshold garch model for the french franc/deutschmark exchange rate', *Journal of Forecasting* **20**, 135–143.

Chappel, D., Padmore, J., Mistry, P. & C., E. (1996), 'A threshold model for the french franc/deutschmark exchange rate', *Journal of Forecasting* **15**, 155–164.

Chatterjee, S. & Laudato, M. (1997), 'Genetic algorithms in statistics: procedures and applications', *Communications in Statistics - Theory and Methods* **26(4)**, 1617–1630.

Chen, C. W. S., Cherng, T.-H. & Wu, B. (2001), 'On the selection of subset bilinear time series models: a genetic algorithm approach', *Computational statistics* **16**, 505–517.

De Jong, K. A. (1975), An analysis of the behavior of a class of genetic adaptive systems, Ph.d. thesis, Dept. of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI.

Engle, R. F. (1982), 'Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation', *Econometrica* **50**, 987–1007.

Feucht, M., Muller, U., Witte, H., Schmidt, K., Arnold, M., Benninger, F., Steinberger, K. & Friedrich, M. H. (1998), 'Nonlinear dynamic of 3 hz spike-and-wave discharges recorded during typical absence seizures in children', *Cerebral Cortex* **8**, 524–533.

Franses, P. H. & van Dijk, D. (2000), *Nonlinear Time Series Models in Empirical Finance*, Cambridge University Press, Cambridge.

Goldberg, D. E. (1989), *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading, MA.

Granger, C. W. J. & Terasvirta, T. (1993), *Modeling Nonlinear Economic Relationships*, Oxford University Press, Oxford.

Haupt, R. L. & Haupt, S. E. (2004), *Practical genetic algorithms (2nd edition)*, John Wiley & Sons, Inc., Hoboken, New Jersey NJ.

Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

Jennison, C. & Sheehan, N. (1995), 'Theoretical and empirical properties of the genetic algorithm as a numerical optimizer', *Journal of Computational and Graphical Statistics* **4**, 296–318.

Li, C. W. & Li, W. K. (1996), 'On a double-threshold autoregressive heteroscedastic time series model', *Journal of Applied Econometrics* **11**, 253–274.

Li, W. K. & Lam, K. (1995), 'Modelling asymmetry in stock returns by threshold autoregressive conditional heteroscedastic model', *The Statistician* **44**, 333–341.

Ling, S. (1999), 'On the probabilistic properties of a double threshold ARMA conditional heteroskedasticity model', *Journal of Applied Probability* **36**, 688–705.

Liu, J., Li, W. K. & Li, C. W. (1997), 'On a threshold autoregression with conditional heteroscedastic variances', *Journal of Statistical Planning and Inference* **62**, 279–300.

Ljung, G. M. & Box, G. E. P. (1978), 'On a measure of lack of fit in time series models', *Journal of Applied Probability* **25**, 553–564.

Man, K. F., Tang, K. S. & Kwong, S. (1999), *Genetic Algorithms: Concepts and Designs*, Springer-Verlag, London.

Montgomery, A., Zarnowitz, V., Tsay, R. S. & Tiao, G. (1998), 'Forecasting the u.s. unemployment rate', *Journal of the American Statistical Association* **93**, 478–493.

Nelson, D. B. & Cao, C. Q. (1992), 'Inequality constraints in the univariate GARCH model', *Journal of Business & Economic Statistics* **10**, 229–235.

Pasia, J. M., Hermosilla, A. Y. & Ombao, H. (2005), 'A useful tool for statistical estimation: genetic algorithms', *Journal of Statistical Computation and Simulation* **75**, 237–251.

Pirlot, M. (1996), 'General local search methods', *European Journal of Operational Research* **92**, 493–511.

Priestley, M. B. (1988), *Non-linear and Non-stationary Time Series Analysis*, Academic Press.

Prugel-Bennett, A. & Shapiro, J. L. (1994), 'An analysis of genetic algorithms using statistical mechanics', *Physical Review Letters* **72**, 1305–1309.

Reeves, C. R. & Rowe, J. E. (2003), *Genetic algorithms - Principles and Perspective: A Guide to GA Theory*, Kluwer Academic Publishers, London.

Tong, H. (1978), On a threshold model, *in* C. H. Chen, ed., 'Pattern Recognition and Signal Processing', Sijhoff and Noordhoff, Amsterdam, pp. 101–141.

Tong, H. (1990), *Non Linear Time Series: A Dynamical System Approach*, Oxford University Press, Oxford.

Tong, H. & Yeung, I. (1991), 'On tests for self-exciting threshold autoregressive-type non-linearity in partially observed time series', *Applied Statistics* **40**, 43–62.

Tsay, R. S. (1989), 'Testing and modeling threshold autoregressive processes', *Journal of the American Statistical Association* **84**, 231–240.

van Dijk, D., Terasvirta, T. & Franses, P. H. (2002), 'Smooth transition autoregressive models - a survey of recent developments', *Econometric Reviews* **21**, 1–47.

Vose, M. D. (1999), *The Simple Genetic Algorithm: Foundations and Theory*, The MIT Press, Cambridge, Massachusetts.

Watier, L. & Richardson, S. (1995), 'Modelling of an epidemiological time series by a threshold autoregressive model', *Journal of the Royal Statistical Society* **44**, 353–364.

Winker, P. (2001), *Optimization heuristics in econometrics: applications of threshold accepting*, Wiley, Chichester.

Winker, P. & Gilli, M. (2004), 'Applications of optimization heuristics to estimation and modelling problems', *Computational Statistics & Data Analysis* **47**, 211–223.

Wu, B. & Chang, C.-L. (2002), 'Using genetic algorithms to parameters (d,r) estimation for threshold autoregressive models', *Computational Statistics & Data Analysis* **38**, 315–330.

Yadav, P. K., Pope, P. F. & Paudyal, K. (1994), 'Threshold autoregressive modelling in finance: The price difference of equivalent assets', *Mathematical Finance* **4**, 205–221.